How Multi-Timescale Memory Cools Optimization: Theory and Evidence from AdEMAMix

Anonymous authors
Paper under double-blind review

Abstract

Augmenting AdamW with multiple momentum timescales improves large language model pretraining, yet the mechanism driving these gains remains poorly understood. We identify an interaction between multi-timescale optimizer memory and the geometry of normalized networks: memory changes parameter norms and thereby controls how fast parameters rotate. Starting from exact norm dynamics under decoupled weight decay, we show that the parameter norm depends on the temporal correlation of the updates, with weight decay controlling how long these correlations influence optimization. We show how an independently weighted slow momentum can substantially increase the norm of parameters, while contributing negligible power to the instantaneous update, therefore acting as an *endogenous* annealing schedule. We study this mechanism in AdEMAMix (Pagliardini et al., 2025) for language-model pretraining. Controlled interventions show that this scheduling mechanism has clear practical consequences and is an important factor behind AdEMAMix's advantage over AdamW.

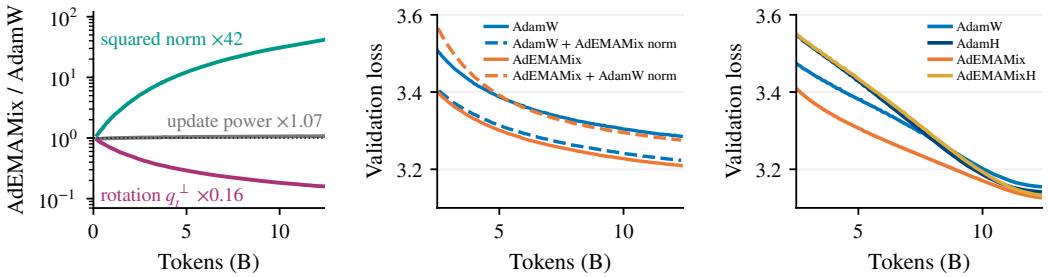
1 Introduction

The rapid scaling of large language models (LLMs) has increasingly relied on practical optimizer heuristics, while a principled understanding of the underlying learning mechanics (Simon et al., 2026) remains an open challenge. Multi-timescale optimizer memory is a recent example. While AdamW (Loshchilov & Hutter, 2019) has long served as the standard optimizer, recent work demonstrates that augmenting it with multiple memory timescales yields significant, consistent gains across large-scale pretraining regimes (Pagliardini et al., 2025; Semenov et al., 2025; Hernández-Cano et al., 2026; Ferbach et al., 2026; Everett & Qiu, 2026). Most notably, AdEMAMix (Pagliardini et al., 2025) equips AdamW with a second, much slower independently weighted exponential moving average (EMA) of past gradients whose timescale and strength evolve over training. Despite recent work relating AdEMAMix to accelerated stochastic methods (Morwani et al., 2025; Ferbach et al., 2026), these perspectives do not explain how its multi-timescale memory alters the training dynamics of modern neural networks. Why such a slow momentum channel consistently benefits large-scale pretraining is therefore an open puzzle.

In this work, we show that a major driver of AdEMAMix's success is an indirect, geometric mechanism: *the interaction between optimizer memory and parameter scale invariance*. Modern language models rely heavily on normalization layers, making parameter groups exactly or approximately scale-invariant: the loss depends only on parameter directions rather than also on their norms (van Laarhoven, 2017; Li & Arora, 2020). However, the norm remains dynamically critical: it interacts with the learning rate (LR) and with optimizer updates to regulate the parameters' rotation speed during training (Hoffer et al., 2018; Wan et al., 2021; Kosson et al., 2024). We study how an optimizer's memory configuration governs this radial norm trajectory and, in turn, endogenously regulates the parameter rotational dynamics:

How does multi-timescale optimizer memory shape the parameter norm, and how does this transform learning-rate schedules into endogenous rotational schedules?

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) (b) (c)

Figure 1: Same learning rate and weight decay, different memory, in 124M Llama pretraining. (a) AdEMAMix/AdamW ratios at the same constant LR: update power stays within 10%, while the squared parameter norm grows $42\times$ and tangential rotation slows about sixfold. (b) Norm transplants, each run at its best constant LR. AdamW with AdEMAMix's norm recovers most of the final-loss gap; AdEMAMix with AdamW's instead worsens. (c) Validation loss under cosine LR with a free radius and with Hyperball extension (AdamH, AdEMAMixH): fixing the radius helps AdamW but hurts AdEMAMix, which collapses on top of AdamH.

Prior work establishes that momentum can alter scale-invariant norms and rotational dynamics (Heo et al., 2021; Wan et al., 2021; Kosson et al., 2024). Our focus is on how the *temporal structure and timescales* of optimizer memory determine this norm trajectory. The key mechanism is the separation between what an optimizer momentum buffer contributes to the update applied now versus what its correlated history leaves behind in the parameter norm. Starting from the exact norm dynamics under decoupled weight decay (WD), we show temporal correlations of the update govern the norm at equilibrium, with WD weighting correlations between updates at different steps. This separates *instantaneous update power* from the contribution of *temporal correlation* to the norm response: a memory channel can contribute little instantaneous power while its correlated history exerts a much larger accumulated effect on the norm, thereby dampening future rotation. Multi-timescale memory strongly amplifies this separation. In particular, an independently weighted slow channel can make this correlation contribution large without noticeably increasing instantaneous update power, accumulating a disproportionate norm contribution that anneals rotational motion. AdEMAMix provides a clear instance of this mechanism: in our language-model runs, its slow, independently weighted, and scheduled momentum branch contributes only a few percent of instantaneous update power, yet its correlated history has a disproportionately persistent effect. The parameter norm in scale-invariant groups becomes roughly six times larger than AdamW's under the same nominal LR and WD, while rotational motion is reduced by about the same factor (Fig. 1a). The norm trajectory induced by optimizer memory is therefore not merely a geometric side effect: it transforms the nominal LR schedule into an optimizer-specific rotational schedule, making optimizer memory an integral part of the scheduling problem itself.

To quantify the optimization consequences of this transformation, we evaluate the resulting dynamics through a finite-horizon scale-invariant scheduling criterion (Anonymous authors, 2026), building on the observation of Schaipp et al. (2025) that convex last-iterate bounds capture several empirical properties of LR scheduling. Holding the nominal schedule fixed (e.g., cosine (Loshchilov & Hutter, 2017) or a warmup-stable-decay (WSD) schedule (Hu et al., 2024; Hägele et al., 2024)), changing optimizer memory can materially alter the resulting rotational schedule and its finite-horizon criterion. We use this criterion as a simulation of the scheduling effect, comparing prescribed memory policies under independently tuned nominal learning rates.

We next ask whether this channel is causally important in language-model training. Recent methods eliminate this radial degree of freedom by constraining optimization to a fixed-radius sphere (Wen et al., 2026; Hägele et al., 2026; Hübler et al., 2026). Recent work provides complementary motivation for studying realized rotational trajectories, showing that matching effective or angular step-size dynamics can reproduce much of the behavior of otherwise different optimizers or hyperparameter settings (Liu et al., 2026; Yuan et al., 2026; Xiao et al., 2026). Controlled interventions indicate that the norm channel is important for AdEMAMix. Removing the optimizer-induced radial component collapses AdEMAMix's norm and degrades performance. Transplanting AdEMAMix's

norm trajectory to AdamW markedly improves AdamW's performance and shifts its LR operating point, while imposing AdamW's norm trajectory on AdEMAMix degrades it despite full LR retuning (Fig. 1b). Finally, under a fixed-radius Hyperball control (Wen et al., 2026), fixing the radius helps AdamW but hurts AdEMAMix, and the benefit of slow momentum shrinks to about a third (Fig. 1c). Together, these results indicate that norm-induced rotational scheduling is a major mediator of AdEMAMix's advantage over AdamW.

More broadly, our results show that the LR, WD, and optimizer memory cannot be understood independently for scale-invariant parameters: together they generate the rotational schedule actually executed during training. We therefore recognize that this realized rotational schedule, rather than any individual hyperparameter or schedule in isolation, is a natural object for understanding and comparing optimizer dynamics and performance.

Our Contributions Our main contributions are:

- **Norm dynamics from optimizer memory.** We develop a framework that expresses radial dynamics through the temporal structure of the update, separating instantaneous power from the accumulated contribution of correlated updates.
- **Rotational slowdown from multi-timescale memory.** This decomposition exposes how different memory architectures, and in particular independently weighted multiple timescales, can generate outsized radial effects while contributing little instantaneous power, thereby increasing the parameter norm and annealing future rotational motion.
- **Rotational scheduling and empirical validation.** We connect these dynamics to finite-horizon rotational scheduling and show through controlled LLM interventions that the norm channel explains a large part of AdEMAMix's advantage over AdamW.

App. A discusses related work on multi-timescale optimizer memory, scale-invariant geometry, and radial dynamics.

2 Background: AdEMAMix and Scale-Invariant Dynamics

2.1 AdEMAMix

We consider updates with decoupled weight decay,

$$\boldsymbol{x}_{t+1} = \delta_t \boldsymbol{x}_t - \eta_t \boldsymbol{u}_t, \qquad \delta_t := 1 - \eta_t \lambda_t \in (0, 1], \tag{2.1}$$

where $\boldsymbol{x}_t \in \mathbb{R}^d$, $\eta_t > 0$ is the learning rate, $\lambda_t \geq 0$ is the weight-decay coefficient, and $\boldsymbol{u}_t$ is the optimizer update.

Definition 2.1 (AdEMAMix Update)**.** Given the current gradient $\boldsymbol{g}_t$, AdEMAMix (Pagliardini et al., 2025) combines fast and slow momentum buffers with a shared second-moment scale:

$$\begin{aligned} \boldsymbol{m}_{t+1}^{(j)} &= \beta_t^{(j)} \boldsymbol{m}_t^{(j)} + (1 - \beta_t^{(j)}) \boldsymbol{g}_t, \qquad j \in \{f, s\}, \\ \boldsymbol{v}_{t+1} &= \beta_2 \boldsymbol{v}_t + (1 - \beta_2) \boldsymbol{g}_t^2, \qquad \boldsymbol{u}_t = \frac{\boldsymbol{m}_{t+1}^{(f)} + \alpha_t \boldsymbol{m}_{t+1}^{(s)}}{\sqrt{\boldsymbol{v}_{t+1}} + \varepsilon}. \end{aligned} \tag{2.2}$$

We have that $0 \leq \beta_t^{(f)} = \beta^{(f)}, \beta_t^{(s)}, \beta_2 < 1$, $\alpha_t \geq 0$, and $\varepsilon > 0$. AdamW (Loshchilov & Hutter, 2019) is the case $\alpha_t = 0$. α_t and $\beta_t^{(s)}$ are warmed up through training (App. F.1). For simplicity we ignore bias correction.

2.2 Scale Invariance, Norms, and Rotational Motion

A nonzero parameter group is scale-invariant if $L(\kappa \boldsymbol{x}) = L(\boldsymbol{x})$ for $\kappa > 0$, so the loss depends only on its unit direction $\hat{\boldsymbol{x}} := \boldsymbol{x} / \|\boldsymbol{x}\|$. Differentiating this identity in κ and in $\boldsymbol{x}$, respectively, gives

$$\langle \nabla L(\boldsymbol{x}), \boldsymbol{x} \rangle = 0, \qquad \nabla L(\kappa \boldsymbol{x}) = \kappa^{-1} \nabla L(\boldsymbol{x}).$$

Thus the gradient is tangent to the weight sphere and scales inversely with radius (Kosson et al., 2024; Arora et al., 2019). Equivalently, $\nabla L(\boldsymbol{x}) = \|\boldsymbol{x}\|^{-1} \nabla_{\mathbb{S}} L(\hat{\boldsymbol{x}})$, where $\nabla_{\mathbb{S}} L(\hat{\boldsymbol{x}}) := (\boldsymbol{I} - \hat{\boldsymbol{x}}\hat{\boldsymbol{x}}^\top) \nabla L(\hat{\boldsymbol{x}})$ is the spherical gradient; App. B gives the derivations.

Directional motion depends on the update's radial and tangential components, $u_{t,\parallel} := \langle \boldsymbol{u}_t, \hat{\boldsymbol{x}}_t \rangle$ and $\boldsymbol{u}_{t,\perp} := \boldsymbol{u}_t - u_{t,\parallel}\hat{\boldsymbol{x}}_t$. Dividing their displacements by the radius gives

$$q^{\parallel}{}_t := \frac{\eta_t u_{t,\parallel}}{\|\boldsymbol{x}_t\|}, \qquad q^{\perp}{}_t := \frac{\eta_t \|\boldsymbol{u}_{t,\perp}\|}{\|\boldsymbol{x}_t\|}.$$

The total relative update is $q^{\mathrm{tot}}{}_t := \eta_t \|\boldsymbol{u}_t\| / \|\boldsymbol{x}_t\|$. The turning angle $\Delta\phi_t := \angle(\boldsymbol{x}_t, \boldsymbol{x}_{t+1})$ satisfies $\Delta\phi_t = \arctan\big(q^{\perp}{}_t/(\delta_t - q^{\parallel}{}_t)\big)$ when $\delta_t - q^{\parallel}{}_t > 0$. For small relative updates and small weight decay, we can approximate $q^{\perp}{}_t \simeq \Delta\phi_t$ (Kosson et al., 2024); App. B.2 gives the exact relation and approximation.

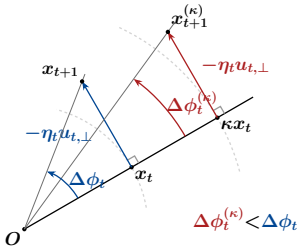


Figure 2: Equal tangential updates produce smaller rotations at larger radii.

Takeaway

For a fixed tangential displacement, a larger parameter norm gives a smaller rotation.

3 The Empirical Puzzle

AdEMAMix generates roughly sixfold larger parameter norms and fivefold slower rotation than AdamW, although the slow buffer's own contribution to instantaneous update power is negligible. We observe this contrast in 124M-parameter Llama pretraining on FineWeb, with the same learning-rate schedule, peak rate $8 \cdot 10^{-4}$, and weight decay for both optimizers.

Fig. 1a shows the norm increase under a constant learning rate; Fig. 10 in App. F.2 shows the norm and rotation trajectories under both constant and cosine rates. Here and below, the matrix norm is the aggregate $\|\boldsymbol{x}_t\| := (\sum_m \|\boldsymbol{x}_{m,t}\|_F^2)^{1/2}$ over the hidden matrices that are weight-decayed, and specified in App. F.1. The larger radius accompanies much smaller tangential relative updates.

To compare this effect with instantaneous power, let $\boldsymbol{u}_t^{(f)}$ and $\boldsymbol{u}_t^{(s)}$ denote the preconditioned fast and slow buffers. Then $\boldsymbol{u}_t = \boldsymbol{u}_t^{(f)} + \alpha_t \boldsymbol{u}_t^{(s)}$ and

$$\|\boldsymbol{u}_t\|^2 = \left\|\boldsymbol{u}_t^{(f)}\right\|^2 + 2\alpha_t \left\langle \boldsymbol{u}_t^{(f)}, \boldsymbol{u}_t^{(s)} \right\rangle + \alpha_t^2 \left\|\boldsymbol{u}_t^{(s)}\right\|^2. \tag{3.1}$$

Our experiments show that the last two terms occupy only a small share of the total power: over the final 30% of training, the three terms account for 95.0%, 2.0%, and 3.0% under constant LR, and 93.7%, 2.2%, and 4.1% under cosine, respectively (App. F.2). So a question arises: *how can this small instantaneous contribution accompany such a large norm increase and rotational slowdown?*

4 Memory-Induced Norm Cooling

4.1 Temporal Correlations Control the Parameter Norm

To gain a better understanding of AdEMAMix, we start by expanding the parameter norm recursion.

Lemma 4.1 (Squared-Norm Growth Identity)**.** *Every update in* (2.1) *satisfies*

$$\|\boldsymbol{x}_{t+1}\|^2 = \delta_t^2 \|\boldsymbol{x}_t\|^2 - 2\delta_t \eta_t \langle \boldsymbol{x}_t, \boldsymbol{u}_t \rangle + \eta_t^2 \|\boldsymbol{u}_t\|^2. \tag{4.1}$$

The first term contracts the previous squared radius thanks to WD. The second is signed radial contribution: alignment of $\boldsymbol{u}_t$ against $\boldsymbol{x}_t$ drives outward motion. The last term is instantaneous displacement power. Without decay, and for heavy-ball momentum on tangent gradients, the radial-work term equals the accumulated past-update power through which momentum grows scale-invariant norms in Heo et al. (2021, Lemma 2.2).

Definition 4.2 (Update Correlations and Their Norm Contribution)**.** For updates $(\boldsymbol{u}_t)_{t\in\mathbb{Z}}$ with finite second moments, constant mean, and $C_u(h) := \mathbb{E}\,\langle \boldsymbol{u}_t, \boldsymbol{u}_{t-h}\rangle$ independent of t, define, for $0 < \delta < 1$,

$$\mathcal{C}_\delta[\boldsymbol{u}] := C_u(0) + 2\sum_{h\geq 1} \delta^h C_u(h). \tag{4.2}$$

This combines instantaneous update power $C_u(0) = \mathbb{E}\,\|\boldsymbol{u}_t\|^2$ with correlations between updates h steps apart, weighted by δ.

Theorem 4.3 (Stationary Norm and Relative-Update Scale)**.** *Let the updates satisfy Definition 4.2 and the parameter obey* (2.1) *for all $t \in \mathbb{Z}$, with constant $\eta > 0$ and $0 < \delta < 1$. Assume radius stationarity: $\mathbb{E}\,\|\boldsymbol{x}_t\|^2 = \bar{R}^2 < \infty$ is independent of t. For $C_u(0) > 0$, define $q_{\bar{R}} := \eta\sqrt{\mathbb{E}\,\|\boldsymbol{u}_t\|^2}/\bar{R}$. Then*

$$\bar{R}^2 = \frac{\eta^2 \mathcal{C}_\delta[\boldsymbol{u}]}{1-\delta^2}, \qquad q_{\bar{R}}^2 = (1-\delta^2)\frac{C_u(0)}{\mathcal{C}_\delta[\boldsymbol{u}]}. \tag{4.3}$$

Positive temporal correlations $\mathcal{C}_\delta[\boldsymbol{u}]$ can therefore increase the radius and reduce the relative-update scale even when instantaneous power $C_u(0)$ changes little. For small steps and weak decay $q_{\bar{R}} \approx \sqrt{\mathbb{E}[q_t^2]}$. The proof is in App. C.1. Under the isotropic Gaussian-gradient model of Wen et al. (2026), where the update correlations have closed forms, our radius identity reduces to their equilibrium (Wen et al., 2026, Eq. 39).

4.2 Optimizer Memory Generates Temporal Correlation

We are now interested in understanding the effect of the optimizer's memory structure onto norm and rotational dynamics. For this, we model an optimizer update as a linear combination of its past inputs:

$$\widehat{\boldsymbol{u}}_t = \sum_{k\geq 0} e_k \boldsymbol{a}_{t-k}. \tag{4.4}$$

The execution kernel e_k is the weight given to an input k steps old. This form uses constant coefficients and an infinite history. The following choices recover familiar optimizers:

Optimizer	Input $\boldsymbol{a}_t$	Execution kernel
Gradient descent / SGD	Full / stochastic gradient	$e_k^{\mathrm{GD}} = e_k^{\mathrm{SGD}} = \mathbf{1}_{k=0}$
Momentum SGD	Stochastic gradient	$e_k^{\mathrm{SGDm}} = (1-\beta)\beta^k$
AdamW	$\boldsymbol{g}_t/(\sqrt{\boldsymbol{v}_{t+1}}+\varepsilon)$	$e_k^{\mathrm{AdamW}} = (1-\beta^{(f)})(\beta^{(f)})^k$
AdEMAMix	$\boldsymbol{g}_t/(\sqrt{\boldsymbol{v}_{t+1}}+\varepsilon)$	$e_k^{\mathrm{AMix}} = e_k^{\mathrm{AdamW}} + \alpha(1-\beta^{(s)})(\beta^{(s)})^k$

For the raw-gradient methods, the representation is exact, using the available history for zero-start momentum. For AdamW and AdEMAMix, it fixes each input's divisor at the time that input enters the buffer, whereas the optimizer divides by the current divisor. This approximation is reminiscent of the common slowly-varying preconditioner approximation (e.g. (Kingma & Ba, 2014)).

To isolate temporal correlations generated by memory, we adopt the noise-dominated setup of Kosson et al. (2024), whose random-walk analysis with gradients uncorrelated across steps predicts AdamW's equilibrium norm and rotation in trained networks.

Assumption 4.4 (Zero-Mean, Uncorrelated Inputs)**.** The inputs satisfy $\mathbb{E}\boldsymbol{a}_t = \mathbf{0}$, $\mathbb{E}\,\|\boldsymbol{a}_t\|^2 = C_a(0) < \infty$ for every t, and $\mathbb{E}\,\langle \boldsymbol{a}_t, \boldsymbol{a}_r\rangle = 0$ for $t \neq r$.

Definition 4.5 (Kernel Power and Norm Factors)**.** For a real, absolutely summable kernel e, define

$$\Gamma_e(h) := \sum_{n\geq 0} e_n e_{n+h}, \qquad P[e] := \Gamma_e(0), \qquad F_\delta[e] := P[e] + 2\sum_{h\geq 1}\delta^h \Gamma_e(h), \tag{4.5}$$

with $0 \leq \delta \leq 1$. Here $P[e]$ measures current power; $F_\delta[e]$ also includes the temporal correlations weighted by weight decay.

Under Assumption 4.4, $C_{\widehat{u}}(h) = C_a(0)\Gamma_e(h)$, so modeled update power is $C_a(0)P[e]$ and its norm contribution is $\mathcal{C}_\delta[\widehat{\boldsymbol{u}}] = C_a(0)F_\delta[e]$.

Proposition 4.6 (Stationary Norm Under Uncorrelated Inputs). *Under Assumption 4.4, let e be absolutely summable and let* (2.1) *be driven by $\widehat{\boldsymbol{u}}_t$. Assume radius stationarity, $\mathbb{E}\,\|\boldsymbol{x}_t\|^2 = \bar{R}^2 < \infty$ independent of t, with constant $\eta > 0$ and $0 < \delta < 1$. Then*

$$\bar{R}^2 = \frac{\eta^2 C_a(0) F_\delta[e]}{1-\delta^2}, \qquad q_{\bar{R}}^2 = (1-\delta^2)\frac{P[e]}{F_\delta[e]} \quad \big(C_a(0)P[e] > 0\big). \tag{4.6}$$

The proof is in App. C.2. Extending to inputs with a nonzero constant mean is straightforward: center the inputs and add the mean contribution, as derived in App. C.2.

Radius comparisons require specifying how input power changes between optimizers. For AdamW and AdEMAMix, the normalization of preconditioning cancels a common gradient rescaling when the divisor scales with it, motivating approximately radius-independent input power (Kosson et al., 2024). The live-preconditioned multilayer perceptron (MLP) in App. D.2 illustrates the normalized-input model's norm responses.

Corollary 4.7 (One Unit-Gain EMA). *For $e_k^{\mathrm{AdamW}} = (1-\beta)\beta^k$, $0 \le \beta < 1$, and $0 \le \delta \le 1$,*

$$P[e^{\mathrm{AdamW}}] = \frac{1-\beta}{1+\beta}, \quad F_\delta[e^{\mathrm{AdamW}}] = \frac{1-\beta}{1+\beta}\frac{1+\beta\delta}{1-\beta\delta}, \quad \frac{P[e^{\mathrm{AdamW}}]}{F_\delta[e^{\mathrm{AdamW}}]} = \frac{1-\beta\delta}{1+\beta\delta}. \tag{4.7}$$

For $0 < \delta < 1$, increasing β decreases both power and the norm factor, with power decreasing proportionally faster: $\partial_\beta \log P < \partial_\beta \log F_\delta < 0$.

At fixed LR, decay, and input power, longer one-buffer memory thus lowers both the modeled radius and the relative-update scale. As $\delta \to 1^-$, $F_\delta[e]$ tends to the squared kernel mass $(\sum_k e_k)^2$, which equals one for every unit-gain EMA; this limit recovers the total-update-contribution construction of Kosson et al. (2024, Secs. 3.1–3.2), which assigns each input's full future contribution to the step at which it enters. For short memory relative to weight decay, $1-\delta \ll 1-\beta$, $F_\delta \approx 1$ recovers their β-independent AdamW radius. Instead, for longer memory, $1-\delta \approx 1-\beta$, F_δ lies below the squared mass for every nonnegative kernel. At $\delta = 1$ and matched update power, $F_1/P = (1+\beta)/(1-\beta)$ is the asymptotic norm-growth factor of momentum without decay (Heo et al., 2021, Cor. 2.3); decay tempers it to $(1+\beta\delta)/(1-\beta\delta)$. Derivatives and the gain calculation are in App. C.3.

4.3 The Two-Timescale AdEMAMix Channel

The kernel e^{AMix} adds slow memory while preserving the fast buffer. Unlike a gain change, e.g., increasing η, its longer temporal correlations can increase the norm factor proportionally faster than the current power.

Theorem 4.8 (Two-Channel Power and Temporal Correlation). *Fix $0 < \delta < 1$, $0 \le \beta^{(f)} < \beta^{(s)} < 1$, and $\alpha \ge 0$. Both $P[e^{\mathrm{AMix}}]$ and $F_\delta[e^{\mathrm{AMix}}]$ increase strictly with α, while*

$$\frac{\mathrm{d}}{\mathrm{d}\alpha}\frac{P[e^{\mathrm{AMix}}]}{F_\delta[e^{\mathrm{AMix}}]} < 0. \tag{4.8}$$

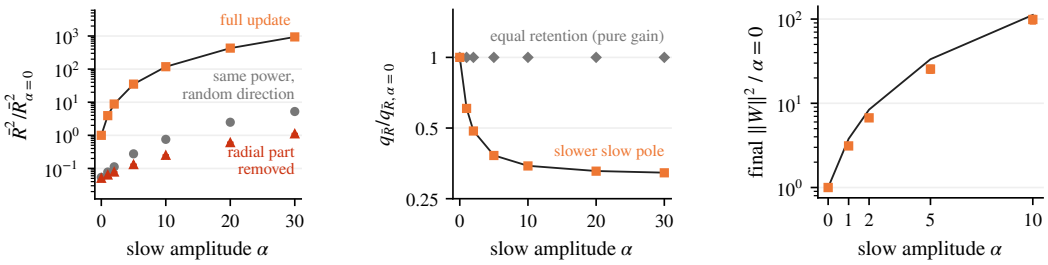
The exact formulas and proof are in App. C.4. Under Proposition 4.6, at matched input power, LR, and decay, *the modeled radius increases while the relative-update scale decreases*.

Example: Little Power, Substantial Cooling. For finite fixed α, consider $1-\delta \ll 1-\beta^{(s)} \ll 1-\beta^{(f)}$: slow memory is much longer than fast memory, but both are short relative to weight decay. The pairwise formulas give

$$P[e^{\mathrm{AMix}}] \approx P[e^{\mathrm{AdamW}}] + \left(2\alpha + \frac{\alpha^2}{2}\right)(1-\beta^{(s)}), \qquad F_\delta[e^{\mathrm{AMix}}] \approx (1+\alpha)^2.$$

The slow buffer spreads its power across many steps, while weak decay preserves its temporal correlations. If $(2\alpha + \alpha^2/2)(1-\beta^{(s)}) \ll P[e^{\mathrm{AdamW}}]$, current power barely changes, yet the modeled radius grows by approximately $1+\alpha$ and the relative-update scale falls by the same factor.

The effect of slow memory is non-monotonic. If α is fixed and the slow memory grows much longer than the weight-decay timescale, $\beta^{(s)} \to 1^-$, then $P[e^{\mathrm{AMix}}]/F_\delta[e^{\mathrm{AMix}}]$ returns to its AdamW



(a) (b) (c)

Figure 3: The norm-cooling predictions in controlled settings; markers are measurements, black lines the model, and all quantities are relative to $\alpha = 0$. (a) Tangent-noise random walk with $\beta^{(s)} = 0.99$: squared radius for the full update, for the same update power with randomized directions, and for the update with its radial part removed. (b) Relative-update scale $q_{\bar{R}}$ for the same sweep, and for an equal-retention control $\beta^{(s)} = \beta^{(f)}$ that changes only gain. (c) Final squared norm of the scale-invariant MLP hidden matrices (geometric mean over matrices, five seeds) against the finite-history model; the model's scale is calibrated at $\alpha = 0$ and cancels in the ratio.

value: weight decay shrinks old updates before the slow buffer reuses them (App. C.5). Our LLM runs are near this regime: weight decay removes about half of the norm increase that slow memory would otherwise give.

Fig. 3 simulates these predictions. In a tangent-noise random walk, Gaussian inputs are projected onto the current weight sphere, and the parameter radius evolves under momentum and decay (App. D.1). Raising α grows the radius as predicted, while the same update power with randomized directions, or the update without its radial part, grows it far less (Fig. 3a). The relative-update scale falls with α, while $\beta^{(s)} = \beta^{(f)}$ changes only the gain and leaves the curve flat (Fig. 3b).

Takeaway

In the stationary uncorrelated-input model, slow memory can raise the parameter norm substantially with little added update power at matched input power, learning rate, and decay. The larger norm reduces rotational motion.

4.4 Finite-Time Radius and Rotational Schedules

For scheduled training, the finite model gives a radius trajectory and a time-varying update-power factor. Write $\widehat{\boldsymbol{u}}_t = \sum_{i=0}^{t} E_{t,i}\boldsymbol{a}_i$, where $E_{t,i}$ is the weight assigned to input i at step t. Under Assumption 4.4 with fixed input power, let $\boldsymbol{x}_t^{\mathrm{rep}}$ follow (2.1) driven by $\widehat{\boldsymbol{u}}_t$, and define $R_t := \sqrt{\mathbb{E}\,\|\boldsymbol{x}_t^{\mathrm{rep}}\|^2}$ and $P_t := \sum_{i=0}^{t} E_{t,i}^2$. The radius accumulates past updates, while P_t sets current power.

To isolate this scalar scheduling effect, supply R_t to a fresh tangent proposal scaled by $\sqrt{P_t}$. Its rotational coefficient is

$$s_t = \frac{\eta_t \sqrt{P_t}}{\delta_t R_t}. \tag{4.9}$$

Thus radius growth can lower the coefficient even at constant LR. The finite radius recursion and proposal derivation are in App. E.2.

The modeled coefficients reproduce the endogenous cooling of the measured LLM angles, with differences in shape and timing, particularly for constant-rate AdamW (Fig. 7 in App. E.3). Sec. 5.3 tests the same model against a controlled intervention.

4.5 Endogenous Rotational Scheduling

Norm dynamics make the realized rotational schedule depart from the nominal one (Kosson et al., 2024, Sec. 5.2), and through (4.9) optimizer memory becomes part of that schedule. To score the result, we use a finite-horizon criterion for deterministic rotational coefficients under a fresh tangent oracle on a geodesically convex spherical cap (Anonymous authors, 2026); its optimal allocation approaches linear decay over most of a long horizon (App. E.1). The criterion gives two expectations.

Expectation A: Explicit and Endogenous Cooling Interact. Under a constant rate, radius growth supplies the only decay of the rotational coefficient; under cosine, the nominal rate already decays. Accordingly, prescribed slow memory lowers the best tuned score under constant LR but raises it under cosine (App. E.3), consistent with the interaction between LR decay and weight decay reported by Wen et al. (2026). AdEMAMix's advantage should therefore be larger under constant rates (Sec. 5.1).

Expectation B: The Useful Bare Rate Depends on the Norm Path. A larger radius lowers (4.9), so the LR operating point depends on the norm trajectory (Sec. 5.2).

5 Practical Consequences in LLM Pretraining

We test the scheduling consequences of optimizer memory and the role of the norm trajectory in 124M-parameter Llama pretraining on FineWeb (App. F).

5.1 Schedule Dependence of the Memory Benefit

After LR tuning, AdEMAMix improves on AdamW by 0.076 validation loss under constant rates and 0.029 under cosine. Both schedules show an advantage of about 0.07 to 0.09 through the first half of training; once the cosine rate decays, a significant part of the gap reduces, while the constant-rate advantage persists (Fig. 4a). This is the interaction of explicit and endogenous cooling anticipated by Expectation A.

5.2 Norm-Path Interventions

Transferring a norm trajectory changes the useful LR range. After each step, we rescale each norm to the norm of a donor-optimizer. Therefore, the recipient keeps its step direction and optimizer buffers (App. F.3), but gets forced onto a different norm trajectory.

Under constant rates, with every arm retuned, the loss mostly follows the norm path (Fig. 1b). Giving AdamW the norm of AdEMAMix significantly improves over vanilla AdamW, approaching the performance of AdEMAMix. Under cosine, the result is more nuanced, pointing to the interaction between LR decay and norm dynamics.

Removing only the update's radial component, while retaining weight decay, collapses AdEMAMix's norm (Fig. 4b) and performance becomes even worse than AdamW's.

Finally, we leverage the Hyperball (Wen et al., 2026) update to fix the parameter radii, showing how this greatly reduces the benefit of slow memory (Fig. 4c). We compare single-buffer AdamH with AdEMAMixH under Hyperball. Fixing the radius improves AdamW's final loss but worsens AdEMAMix's (Fig. 1c). Furthermore, we observe that the α operating point of AdEMAMix shifts significantly: the optimal value reduces from $\alpha = 12$ to $\alpha \in \{1, 2, 3\}$.

5.3 Switching Off the Slow Channel

We now show a final, peculiar fingerprint of AdEMAMix's norm dynamics happening when switching off the slow channel abruptly during training, setting $\alpha_t = 0$. At constant LR, the loss first drops and then rebounds as the advantage of slow memory is withdrawn (Fig. 5a). We are able to predict these curves by modeling the norm-relaxation due to the effect of WD. As the norm relaxes, rotation reheats; the finite-time model of Sec. 4.4, simulated with the same switch, predicts this reheating (Fig. 5b).

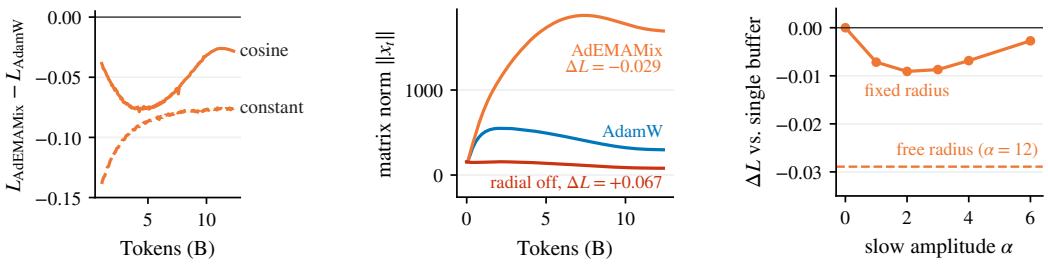


Figure 4: What the norm channel does to the loss. (a) AdEMAMix minus AdamW validation loss at each schedule's tuned rate. (b) Matrix norms at cosine peak rate $8 \cdot 10^{-4}$ with and without the radial update component; labels give final-checkpoint losses relative to AdamW. (c) Fixed-radius HyperBall: final loss of AdEMAMixH relative to single-buffer AdamH across slow amplitudes, against the free-radius gain of AdEMAMix over AdamW (dashed).

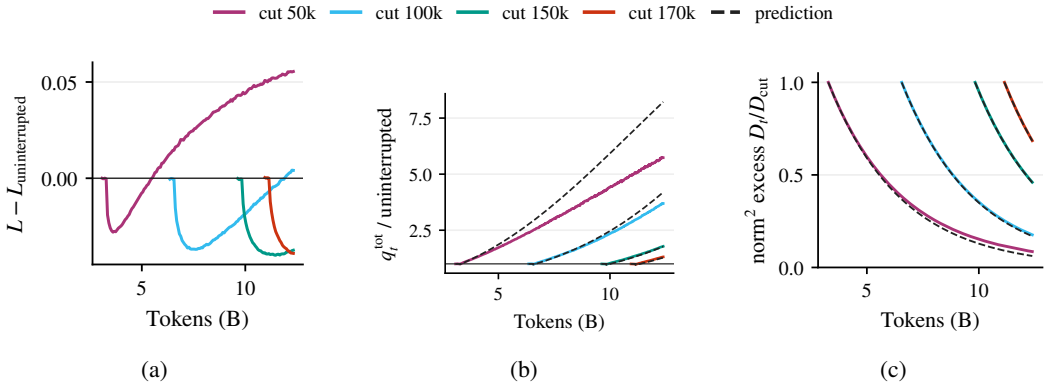


Figure 5: Switching off the slow channel at constant LR 10^{-4}, at four cut times. (a) Validation loss relative to the uninterrupted AdEMAMix run. (b) Total relative update relative to the uninterrupted run; dashed lines are the finite-time model with the same switch. (c) Squared-norm excess over AdamW, normalized at the cut; dashed lines are weight-decay survival predicted by our analysis.

Takeaway

AdEMAMix's norm trajectory is an important factor behind of its success over AdamW.

6 Discussion

Optimizer memory is part of the scheduling problem for scale-invariant parameters. A memory change can alter the norm trajectory and therefore the rotational schedule even when nominal LR and weight decay are held fixed. This suggests choosing memory and LR schedules jointly: additional slow memory may supply useful cooling under constant rates, while an explicitly decaying LR can require a different memory policy. Comparisons between optimizers should therefore examine their realized rotational trajectories alongside nominal hyperparameters.

AI Use Statement

In this work, we used generative AI tools to help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, propose or refine hypotheses, design or provide feedback on research methodology or experiments, assist in writing proofs, implement methods, and interpret results. We did not use generative AI tools

for translation. Generating synthetic datasets, cleaning or reformatting datasets, and qualitative or thematic data analysis were not applicable to this work.

Additionally, we used generative AI tools to create or modify scientific figures or images, create or edit software code, draft parts of the paper, summarize or analyze existing literature, discover research topics or identify gaps, brainstorm, search for information, edit the paper for readability, identify relevant literature, format references, suggest the paper's structure, and propose a title or keywords.

We reviewed all AI-assisted work. We checked LLM-generated research ideas for potential plagiarism through a manual literature survey, verified mathematical proofs by hand, tested LLM-generated code manually, and revised, proofread, improved readability of LLM-generated text. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Anonymous authors. From learning rates to rotational schedules: Finite-horizon theory on the sphere, 2026. Under review at ICLR 2027.

Sanjeev Arora, Zhiyuan Li, and Kaifeng Lyu. Theoretical analysis of auto rate-tuning by batch normalization. In *International Conference on Learning Representations*, 2019. Also arXiv:1812.03981.

Katie Everett and Shikai Qiu. Optimizer memory schedules for outscaling the overtraining axis. *arXiv preprint arXiv:2609.04577*, 2026.

Damien Ferbach, Courtney Paquette, Gauthier Gidel, Katie Everett, and Elliot Paquette. Logarithmic-time schedules for scaling language models with momentum. *arXiv preprint arXiv:2602.05298*, 2026.

Alexander Hägele, Elie Bakouch, Atli Kosson, Loubna Ben Allal, Leandro von Werra, and Martin Jaggi. Scaling laws and compute-optimal training beyond fixed training durations. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Alexander Hägele, Alejandro Hernández-Cano, Atli Kosson, and Martin Jaggi. Improving neural network training by decoupling the magnitude and direction of weight vectors. *arXiv preprint arXiv:2606.25971*, 2026.

Byeongho Heo, Sanghyuk Chun, Seong Joon Oh, Dongyoon Han, Sangdoo Yun, Gyuwan Kim, Youngjung Uh, and Jung-Woo Ha. AdamP: Slowing down the slowdown for momentum optimizers on scale-invariant weights. In *International Conference on Learning Representations*, 2021. arXiv:2006.08217.

Alejandro Hernández-Cano, Alexander Hägele, et al. Apertus: Democratizing open and compliant LLMs for global language environments. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2026. doi: 10.18653/v1/2026.acl-long.2172.

Elad Hoffer, Ron Banner, Itay Golan, and Daniel Soudry. Norm matters: Efficient and accurate normalization schemes in deep networks. In *Advances in Neural Information Processing Systems*, volume 31, 2018. arXiv:1803.01814.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, Xinrong Zhang, Zheng Leng Thai, Kaihuo Zhang, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie Zhou, Jie Cai, Zhongwu Zhai, Ning Ding, Chao Jia, Guoyang Zeng, Dahai Li, Zhiyuan Liu, and Maosong Sun. MiniCPM: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*, 2024.

Florian Hübler, Kai Lion, Antonio Orvieto, and Niao He. Muown implicitly performs angular step-size decay. *arXiv preprint arXiv:2606.23637*, 2026.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Atli Kosson, Bettina Messmer, and Martin Jaggi. Rotational equilibrium: How weight decay balances learning across neural networks. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 25333–25369. PMLR, 2024.

Zhiyuan Li and Sanjeev Arora. An exponential learning rate schedule for deep learning. In *International Conference on Learning Representations*, 2020. arXiv:1910.07454.

Zihan Liu, Ruiheng Zheng, Shaobo Zhang, Changxin Tian, Kunlong Chen, Zhiqiang Zhang, and Lei Wu. Effective learning rate governs loss dynamics in language model pretraining. *arXiv preprint arXiv:2608.24814*, 2026.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017. arXiv:1608.03983.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. arXiv:1711.05101.

James Lucas, Shengyang Sun, Richard Zemel, and Roger Grosse. Aggregated momentum: Stability through passive damping. *arXiv preprint arXiv:1804.00325*, 2018.

Depen Morwani, Nikhil Vyas, Hanlin Zhang, and Sham Kakade. Connections between schedule-free optimizers, ademamix, and accelerated sgd variants. *arXiv preprint arXiv:2502.02431*, 2025.

Matteo Pagliardini, Pierre Ablin, and David Grangier. The AdEMAMix optimizer: Better, faster, older. In *International Conference on Learning Representations*, 2025. arXiv:2409.03137.

Simon Roburin, Yann de Mont-Marin, Andrei Bursuc, Renaud Marlet, Patrick Pérez, and Mathieu Aubry. Spherical perspective on learning with normalization layers. *Neurocomputing*, 487:66–74, 2022. doi: 10.1016/j.neucom.2022.02.021. arXiv:2006.13382.

Fabian Schaipp, Alexander Hägele, Adrien Taylor, Umut Şimşekli, and Francis Bach. The surprising agreement between convex optimization theory and learning-rate scheduling for large model training. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 53267–53294. PMLR, 2025.

Andrei Semenov, Matteo Pagliardini, and Martin Jaggi. Benchmarking optimizers for large language model pretraining. *arXiv preprint arXiv:2509.01440*, 2025.

Jamie Simon, Daniel Kunin, Alexander Atanasov, Enric Boix-Adserà, Blake Bordelon, Jeremy Cohen, Nikhil Ghosh, Florentin Guth, Arthur Jacot, Mason Kamb, Dhruva Karkada, Eric J. Michaud, Berkan Ottlik, and Joseph Turnbull. There will be a scientific theory of deep learning. *arXiv preprint arXiv:2604.21691*, 2026.

Twan van Laarhoven. L2 regularization versus batch and weight normalization. *arXiv preprint arXiv:1706.05350*, 2017.

Ruosi Wan, Zhanxing Zhu, Xiangyu Zhang, and Jian Sun. Spherical motion dynamics: Learning dynamics of normalized neural network using SGD and weight decay. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Kaiyue Wen, Xingyu Dang, Kaifeng Lyu, Tengyu Ma, and Percy Liang. Fantastic pretraining optimizers and where to find them II: Hyperball optimization. *arXiv preprint arXiv:2606.16899*, 2026.

Yihao Xiao, Jialong Sun, Zitian Gao, Zeming Wei, Chutian Wang, Ran Tao, Jiaye Teng, and Bryan Dai. Hyperball may not be a free lunch. *arXiv preprint arXiv:2607.22444*, 2026.

Jinghui Yuan, Hongtao Zhang, Jade Zou, Tianyu Li, Wenjie Zhou, Tianyu He, and Wei Chen. HyperTransfer: Understanding the equivalence between base optimizer and hyperball. *arXiv preprint arXiv:2609.07017*, 2026.

APPENDIX CONTENTS

A Related Work **13**

B Scale-Invariant Geometry and Rotation Measurements **14**

B.1 Gradient Tangency and Inverse-Radius Scaling 14

B.2 Exact Rotation and Relative-Update Proxies 14

C Proofs and Algebra for Memory-Induced Norm Cooling **15**

C.1 The Update Correlation Law 15

C.2 Execution Kernels and the Uncorrelated-Input Model 17

C.3 Single-Timescale Memory 21

C.4 Two-Timescale ADEMAMIX Kernel Algebra 23

C.5 Weight-Decay Contraction Limits Useful Memory 26

D Controlled Validation of the Mechanism **27**

D.1 Noisy-Sphere Testbed and Kernel Tests 27

D.2 Scale-Invariant MLP Bridge 28

E Rotational-Scheduling Details **29**

E.1 The Scheduling Criterion Used in the Main Text 29

E.2 Forward Construction: From Optimizer Memory to a Rotational Coefficient 30

E.3 Bound Evaluation with Prescribed Memory 32

F LLM Experimental Details and Supplementary Results **32**

F.1 Training, Tuning, and Measurement Protocol 32

F.2 Norm, Rotation, and Instantaneous-Power Measurements 34

F.3 Intervention Definitions 36

F.4 Full Curves for the Main Performance Experiments 37

F.5 Approximate Scale Invariance of the Matrix Bucket 39

A Related Work

Multi-Timescale Optimizer Memory. Pagliardini et al. (2025) bring the idea of multi-timescale optimizer memory (Lucas et al., 2018) to large-scale language model pretraining, with AdEMAMix. They motivate their optimizer heuristically: maintaining the optimizer's sensitivity to recent gradients with a fast momentum buffer while also retaining a much longer-horizon memory of past gradients with an independently weighted slow momentum buffer. For improved practical stability, they also rely on scheduling both the slow-momentum's timescale and its independent weighting over the entire training horizon. The strong empirical performance of AdEMAMix has been confirmed by other large-scale pretraining studies (Semenov et al., 2025; Hernández-Cano et al., 2026). Recent work has connected this construction to accelerated stochastic optimization, showing that AdEMAMix closely resembles accelerated stochastic gradient descent (SGD) variants, in which the weight on the fresh gradient is decoupled from the momentum timescale (Morwani et al., 2025). A similar perspective inspires (Ferbach et al., 2026) to propose Adana as a closely related variant of AdEMAMix, augmented with logarithmic-time schedules for momenta and weight decay hyperparameters. These works interpret persistent memory through stochastic acceleration, stability, and scaling. Our work identifies a different mechanism: multi-timescale memory interacts with the geometry of normalized networks, shaping the realized rotational dynamics of the parameters.

Geometry of Scale-Invariant Optimization. Normalization layers make many parameter groups exactly or approximately scale-invariant, meaning that rescaling the norm of the parameters leaves the represented function and loss unchanged. This induces an intrinsic spherical geometry in which parameter directions alone determine the represented function. Early work established that the norm plays a key role in optimization, as it inversely modulates the rotational motion of the parameters (van Laarhoven, 2017; Hoffer et al., 2018; Li & Arora, 2020): larger norms generally correspond to smaller rotational steps. Roburin et al. (2022) formalize first-order optimization through its induced motion on the hypersphere, deriving effective learning rates and directions for methods including momentum SGD and Adam, and showing how radial terms and accumulated momentum enter these quantities. Wan et al. (2021) analyze the corresponding spherical motion dynamics under SGD and momentum SGD with weight decay, using the per-step rotation angle as a direct measure of directional motion. We build on this geometric foundation to study how scale invariance affects the rotational dynamics induced by multi-timescale optimizer memory.

Radial Dynamics and Rotational Equilibrium. The effect of optimizer memory on the parameter norm has been studied in several recent works. In the absence of weight decay Heo et al. (2021) showed that momentum reuses past updates to produce additional norm growth in scale-invariant parameters, and proposed AdamP to project out the radial component of the momentum update to prevent the associated slowdown of directional motion. With weight decay, Kosson et al. (2024) instead showed that stochastic updates and radial contraction can settle into a *rotational equilibrium*, deriving equilibrium norms and angular update rates for several optimizers. For AdamW, their noise-dominated random-walk analysis predicts an equilibrium radius independent of the first-moment hyperparameter $\beta^{(f)}$ in the short-memory regime. Our analysis uncovers the mechanism underlying this cancellation and shows that an independently weighted second momentum timescale can have a much larger effect on the norm than its instantaneous power suggests. Concurrent with our work, Wen et al. (2026) derived a related autocorrelation-weighted stationary radius expression under an idealized isotropic model, together with the corresponding angular step size. They use this analysis to motivate eliminating the radial dynamics by explicitly fixing parameter and update norms and optimizing on a fixed-radius sphere with the Hyperball optimizer. Our perspective instead treats the radial degree of freedom as a dynamical channel to be explained rather than removed.

Our work connects multi-timescale memory with the geometry of normalized networks, linking the stochastic-acceleration perspective to the dynamics of scale-invariant networks. We provide a mechanistic account of how AdEMAMix's slow memory affects training: a memory channel can contribute little to the update applied at the current step while its correlated history leaves a much larger, persistent effect on the parameter norm. The resulting norm growth gradually anneals angular motion, transforming the learning-rate schedule into an optimizer- and memory-dependent rotational schedule. Through controlled interventions we show that this norm channel accounts for a large part of AdEMAMix's advantage over AdamW.

B Scale-Invariant Geometry and Rotation Measurements

B.1 Gradient Tangency and Inverse-Radius Scaling

Let L be differentiable and scale-invariant on a nonzero parameter group:

$$L(\kappa \boldsymbol{x}) = L(\boldsymbol{x}) \qquad (\kappa > 0 \text{ admissible}). \tag{B.1}$$

We record the short tangency argument from Kosson et al. (2024, App. B) and the gradient scaling used later.

Gradient tangency and inverse-radius scaling. Stretching the weight along its own direction cannot change the loss, so the loss has no slope in that direction. Concretely, for every $h > -1$ the point $\boldsymbol{x}+h\boldsymbol{x} = (1+h)\boldsymbol{x}$ is a positive rescaling of $\boldsymbol{x}$, so the invariance (B.1) gives $L(\boldsymbol{x}+h\boldsymbol{x}) = L(\boldsymbol{x})$: the loss is constant along the whole ray through $\boldsymbol{x}$. Its directional derivative along $\boldsymbol{x}$ therefore vanishes at $\boldsymbol{x}$, and by the chain rule that derivative is the inner product we want,

$$0 = \frac{\mathrm{d}}{\mathrm{d}h} L(\boldsymbol{x} + h\boldsymbol{x})\Big|_{h=0} = \langle \nabla L(\boldsymbol{x}), \boldsymbol{x} \rangle \,, \tag{B.2}$$

Differentiating (B.1) in $\boldsymbol{x}$ instead gives $\kappa \nabla L(\kappa \boldsymbol{x}) = \nabla L(\boldsymbol{x})$, hence

$$\nabla L(\kappa \boldsymbol{x}) = \kappa^{-1} \nabla L(\boldsymbol{x}). \tag{B.3}$$

The raw gradient of a scale-invariant loss is homogeneous of degree -1 in the radius (Arora et al., 2019, Lem. 2.2). □

B.2 Exact Rotation and Relative-Update Proxies

Recall the relative updates from Sec. 2.2:

$$q^{\mathrm{tot}}{}_t := \frac{\eta_t \, \|\boldsymbol{u}_t\|}{\|\boldsymbol{x}_t\|}, \qquad q^{\perp}{}_t := \frac{\eta_t \, \|\boldsymbol{u}_{t,\perp}\|}{\|\boldsymbol{x}_t\|}, \qquad q^{\parallel}{}_t := \frac{\eta_t u_{t,\parallel}}{\|\boldsymbol{x}_t\|}. \tag{B.4}$$

The radial and tangential components are orthogonal, giving

$$(q^{\mathrm{tot}}{}_t)^2 = (q^{\parallel}{}_t)^2 + (q^{\perp}{}_t)^2. \tag{B.5}$$

The exact turning angle is

$$\Delta\phi_t = \arccos \frac{\langle \boldsymbol{x}_t, \boldsymbol{x}_{t+1} \rangle}{\|\boldsymbol{x}_t\| \, \|\boldsymbol{x}_{t+1}\|}. \tag{B.6}$$

We now derive the exact relation (B.7) connecting $\Delta\phi_t$ to these relative updates, along with the small-update limit (B.8), so that the approximations implicit in treating relative updates as angular rates are made explicit. The scaled-radius comparison is illustrated in Fig. 2.

The Exact Relation. *Step 1 (decompose the update).* The update in the radial/tangential frame of $\boldsymbol{x}_t$ is $\boldsymbol{u}_t = u_{t,\parallel} \, \hat{\boldsymbol{x}}_t + \boldsymbol{u}_{t,\perp}$.

Step 2 (the next weight in that frame). Substitution into one decoupled step $\boldsymbol{x}_{t+1} = \delta_t \boldsymbol{x}_t - \eta_t \boldsymbol{u}_t$, with $\delta_t = 1 - \eta_t \lambda_t$ and $\boldsymbol{x}_t = \|\boldsymbol{x}_t\| \, \hat{\boldsymbol{x}}_t$, gives

$$\boldsymbol{x}_{t+1} = \left(\delta_t \, \|\boldsymbol{x}_t\| - \eta_t u_{t,\parallel}\right) \hat{\boldsymbol{x}}_t \;-\; \eta_t \, \boldsymbol{u}_{t,\perp}.$$

The bracket is the radial coordinate; $-\eta_t \boldsymbol{u}_{t,\perp}$, of length $\eta_t \, \|\boldsymbol{u}_{t,\perp}\|$, is the tangential coordinate.

Step 3 (read the angle off the two coordinates). $\boldsymbol{x}_t$ points purely along $\hat{\boldsymbol{x}}_t$, so the angle between $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t+1}$ is the angle, in the plane those two directions span, of the vector with radial coordinate $\delta_t \, \|\boldsymbol{x}_t\| - \eta_t u_{t,\parallel}$ and tangential coordinate $\eta_t \, \|\boldsymbol{u}_{t,\perp}\|$. Whenever the radial coordinate is positive, $\delta_t \, \|\boldsymbol{x}_t\| - \eta_t u_{t,\parallel} > 0$, that angle is

$$\Delta\phi_t = \arctan \frac{\eta_t \, \|\boldsymbol{u}_{t,\perp}\|}{\delta_t \, \|\boldsymbol{x}_t\| - \eta_t u_{t,\parallel}}.$$

Step 4 (normalize by the radius). Dividing numerator and denominator by $\|\boldsymbol{x}_t\|$ rewrites them in the relative updates of (B.4), giving the exact relation

$$\tan \Delta\phi_t = \frac{{q^\perp}_t}{\delta_t - {q^\parallel}_t}, \qquad \Delta\phi_t = \arctan \frac{{q^\perp}_t}{\delta_t - {q^\parallel}_t} \qquad (\delta_t - {q^\parallel}_t > 0). \tag{B.7}$$

The condition $\delta_t - {q^\parallel}_t > 0$ makes the radial coordinate positive and holds in the small-update regime used here.

The Small-Update Limit. Suppose the relative update is small, ${q^{\mathrm{tot}}}_t \ll 1$, and the shrink defect is small, $1 - \delta_t = \eta_t \lambda_t \ll 1$. Three steps take (B.7) to (B.8).

Step 1 (the denominator is close to one). The identity $\delta_t = 1 - \eta_t \lambda_t$ and the Pythagorean bound $|{q^\parallel}_t| \le {q^{\mathrm{tot}}}_t$ from (B.5) show that the denominator of (B.7) differs from one by at most the two small quantities assumed above,

$$\delta_t - {q^\parallel}_t = 1 - \eta_t \lambda_t - {q^\parallel}_t = 1 - O(\eta_t \lambda_t + {q^{\mathrm{tot}}}_t).$$

Step 2 (so the arctangent's argument is small). Its numerator obeys ${q^\perp}_t \le {q^{\mathrm{tot}}}_t \ll 1$ by (B.5) as well, and Step 1 keeps its denominator near one, so the whole argument $z_t = {q^\perp}_t / (\delta_t - {q^\parallel}_t)$ is small, and dividing by that denominator gives $z_t = {q^\perp}_t \, [1 + O(\eta_t \lambda_t + {q^{\mathrm{tot}}}_t)]$.

Step 3 (expand the arctangent about zero). For small z the Taylor series $\arctan z = z - z^3/3 + \cdots$ reads $\arctan z = z\,[1 + O(z^2)]$. Applying it to z_t and substituting Step 2 gives (B.8), the second form using ${q^\perp}_t = {q^{\mathrm{tot}}}_t \, \|\boldsymbol{u}_{t,\perp}\| \,/\, \|\boldsymbol{u}_t\|$, which is (B.4) with the common factor $\eta_t / \|\boldsymbol{x}_t\|$ divided out.

The correction bracket $1 + O(\eta_t \lambda_t + {q^{\mathrm{tot}}}_t)$ collects the two effects the steps introduced. Step 1's denominator contributes the shrink defect $\eta_t \lambda_t = 1 - \delta_t$ together with the radial slice ${q^\parallel}_t$; Step 3's expansion contributes a further term of order $({q^\perp}_t)^2$, which the bracket already covers. Explicitly, the approximation derived above is

$$\Delta\phi_t = {q^\perp}_t [1 + O(\eta_t \lambda_t + {q^{\mathrm{tot}}}_t)] = {q^{\mathrm{tot}}}_t \frac{\|\boldsymbol{u}_{t,\perp}\|}{\|\boldsymbol{u}_t\|} [1 + O(\eta_t \lambda_t + {q^{\mathrm{tot}}}_t)]. \tag{B.8}$$

The second form uses a nonzero update; for $\boldsymbol{u}_t = 0$, both the angle and ${q^\perp}_t$ vanish. The error includes the decay contraction, radial displacement, and cubic arctangent correction.

Kosson et al. (2024) use the same exact angle (their Eq. 3) and adopt $\Delta\phi \approx {q^{\mathrm{tot}}}$ as an operational proxy. Equation (B.8) identifies its tangential fraction and finite-step corrections.

C Proofs and Algebra for Memory-Induced Norm Cooling

C.1 The Update Correlation Law

For the constant-coefficient stationary arguments below, write

$$\boldsymbol{x}_{t+1} = \delta \boldsymbol{x}_t - \eta \boldsymbol{u}_t, \qquad \delta = 1 - \eta\lambda \in (0, 1). \tag{C.1}$$

Proof of Lemma 4.1. Expansion of the squared norm gives

$$\begin{aligned} \|\boldsymbol{x}_{t+1}\|^2 &= \|\delta_t \boldsymbol{x}_t - \eta_t \boldsymbol{u}_t\|^2 \\ &= \delta_t^2 \|\boldsymbol{x}_t\|^2 - 2\delta_t \eta_t \langle \boldsymbol{u}_t, \boldsymbol{x}_t \rangle + \eta_t^2 \|\boldsymbol{u}_t\|^2 . \end{aligned} \tag{C.2}$$

Symmetry of the inner product identifies (4.1). □

Finiteness of the Weight-Decay-Weighted Functional. Symmetry of the real inner product gives

$$C_u(-h) = C_u(h). \tag{C.3}$$

The equilibrium sum converges under the same finite-power assumption. Cauchy–Schwarz, stationarity, and the geometric series give

$$\begin{aligned} |C_u(h)| &\le \sqrt{\mathbb{E}\|\boldsymbol{u}_t\|^2} \sqrt{\mathbb{E}\|\boldsymbol{u}_{t-h}\|^2} = C_u(0), \\ \sum_{h \ge 1} \delta^h |C_u(h)| &\le C_u(0) \sum_{h \ge 1} \delta^h = C_u(0) \frac{\delta}{1 - \delta} < \infty. \end{aligned} \tag{C.4}$$

In particular,

$$\sum_{h\geq 1} \delta^h |C_u(h)| < \infty. \tag{C.5}$$

This convergence concerns a scalar series of correlations and makes the functional below finite. It does not require the correlations themselves to decay to zero: the geometric weight-decay factors already make the series absolutely summable. The proof below uses finite unrollings, while the preceding absolute-convergence bound controls the scalar correlation limit.

Proof of Theorem 4.3. Take expectations in the pathwise identity (4.1). All terms are finite: the squared terms by assumption and the cross term by Cauchy–Schwarz. Radius stationarity and update stationarity give

$$(1-\delta^2)\bar{R}^2 = -2\delta\eta \mathbb{E}\left\langle \boldsymbol{x}_t, \boldsymbol{u}_t \right\rangle + \eta^2 C_u(0).$$

It remains to evaluate the radial cross moment. Iterating (C.1) backward through r steps gives the finite unrolling

$$\boldsymbol{x}_t = \delta^r \boldsymbol{x}_{t-r} - \eta \sum_{h=1}^{r} \delta^{h-1} \boldsymbol{u}_{t-h}.$$

Taking its inner product with $\boldsymbol{u}_t$ and then expectation yields, at every finite r,

$$\mathbb{E}\left\langle \boldsymbol{x}_t, \boldsymbol{u}_t \right\rangle = \delta^r \mathbb{E}\left\langle \boldsymbol{x}_{t-r}, \boldsymbol{u}_t \right\rangle - \eta \sum_{h=1}^{r} \delta^{h-1} \mathbb{E}\left\langle \boldsymbol{u}_{t-h}, \boldsymbol{u}_t \right\rangle.$$

By Cauchy–Schwarz, radius stationarity, finite update power, and $0 < \delta < 1$, the remote-state term satisfies

$$|\delta^r \mathbb{E}\left\langle \boldsymbol{x}_{t-r}, \boldsymbol{u}_t \right\rangle| \leq \delta^r \sqrt{\mathbb{E}\left\|\boldsymbol{x}_{t-r}\right\|^2} \sqrt{\mathbb{E}\left\|\boldsymbol{u}_t\right\|^2} = \delta^r \bar{R} \sqrt{C_u(0)} \xrightarrow[r\to\infty]{} 0.$$

Symmetry of the real inner product and the definition of $C_u(h)$ identify $\mathbb{E}\left\langle \boldsymbol{u}_{t-h}, \boldsymbol{u}_t \right\rangle = C_u(h)$. Moreover, (C.4) and $\delta > 0$ give

$$\sum_{h\geq 1} \delta^{h-1} |C_u(h)| = \frac{1}{\delta} \sum_{h\geq 1} \delta^h |C_u(h)| < \infty.$$

Taking $r \to \infty$ in the finite cross-moment identity therefore gives

$$\mathbb{E}\left\langle \boldsymbol{x}_t, \boldsymbol{u}_t \right\rangle = -\eta \sum_{h\geq 1} \delta^{h-1} C_u(h).$$

This corresponds to the stationary projection coefficient of Wen et al. (2026, Lemma 4.4), here for arbitrary update correlations. Substitution into the stationary squared-radius balance yields

$$\begin{aligned}(1-\delta^2)\bar{R}^2 &= \eta^2 C_u(0) + 2\eta^2 \sum_{h\geq 1} \delta^h C_u(h) \\ &= \eta^2 \left[C_u(0) + 2 \sum_{h\geq 1} \delta^h C_u(h) \right] \\ &= \eta^2 \mathcal{C}_\delta[\boldsymbol{u}].\end{aligned} \tag{C.6}$$

Since $1 - \delta^2 > 0$, division gives the squared-radius identity

$$\bar{R}^2 = \frac{\eta^2 \mathcal{C}_\delta[\boldsymbol{u}]}{1-\delta^2}.$$

The radius identity includes the zero-power case $C_u(0) = 0$. If $C_u(0) > 0$, then $\bar{R} > 0$: otherwise $\boldsymbol{x}_t = \boldsymbol{x}_{t+1} = 0$ almost surely, and the update equation would force $\boldsymbol{u}_t = 0$ almost surely. Substituting the radius identity into $q_{\bar{R}}^2 = \eta^2 C_u(0)/\bar{R}^2$ gives the second identity in (4.3). □

Cumulative Weight-Decay Contraction and Unrolled Trajectory.

Definition C.1 (Cumulative Weight-Decay Contraction)**.** For $0 \le i \le j$, set

$$W_{i,j} := \prod_{r=i}^{j-1}(1-\eta_r\lambda_r), \qquad W_{j,j} = 1. \tag{C.7}$$

The finite update history obeys

$$\boldsymbol{x}_T = W_{0,T}\boldsymbol{x}_0 - \sum_{t=0}^{T-1} W_{t+1,T}\eta_t\boldsymbol{u}_t. \tag{C.8}$$

Proof of the unrolling (C.8) *in Definition C.1.* For compactness write $\delta_t = 1-\eta_t\lambda_t$. Iterating the update (2.1) backward from step T,

$$\begin{aligned}
\boldsymbol{x}_T &= \delta_{T-1}\boldsymbol{x}_{T-1} - \eta_{T-1}\boldsymbol{u}_{T-1} \\
&= \delta_{T-1}\big(\delta_{T-2}\boldsymbol{x}_{T-2} - \eta_{T-2}\boldsymbol{u}_{T-2}\big) - \eta_{T-1}\boldsymbol{u}_{T-1} \\
&= \delta_{T-1}\delta_{T-2}\boldsymbol{x}_{T-2} - \delta_{T-1}\eta_{T-2}\boldsymbol{u}_{T-2} - \eta_{T-1}\boldsymbol{u}_{T-1} \\
&= \delta_{T-1}\delta_{T-2}\delta_{T-3}\boldsymbol{x}_{T-3} - \delta_{T-1}\delta_{T-2}\eta_{T-3}\boldsymbol{u}_{T-3} - \delta_{T-1}\eta_{T-2}\boldsymbol{u}_{T-2} - \eta_{T-1}\boldsymbol{u}_{T-1} \\
&= \cdots \\
&= \left[\prod_{r=T-m}^{T-1}\delta_r\right]\boldsymbol{x}_{T-m} \;-\; \sum_{t=T-m}^{T-1}\left[\prod_{r=t+1}^{T-1}\delta_r\right]\eta_t\,\boldsymbol{u}_t \\
&= \left[\prod_{r=0}^{T-1}(1-\eta_r\lambda_r)\right]\boldsymbol{x}_0 \\
&\quad - \sum_{t=0}^{T-1} W_{t+1,T}\eta_t\,\boldsymbol{u}_t.
\end{aligned} \tag{C.9}$$

Each backward substitution peels off one more shrink factor $\delta_r = 1-\eta_r\lambda_r$; at $m = T$ the recursion reaches $\boldsymbol{x}_0$, giving the last line. The bracketed product attached to step t is $\prod_{r=t+1}^{T-1}(1-\eta_r\lambda_r)$, which is $W_{t+1,T}$ of (C.7) at $i = t+1$; the leading product $\prod_{r=0}^{T-1}(1-\eta_r\lambda_r)$ is $W_{0,T}$ at $i = 0$. Collecting them gives (C.8). □

C.2 Execution Kernels and the Uncorrelated-Input Model

Definition C.2 (Execution Kernel)**.** For a first-moment buffer with retentions ρ_r, define

$$E_{t,i} := (1-\rho_i)\prod_{r=i+1}^{t}\rho_r \qquad (0 \le i \le t). \tag{C.10}$$

With zero initialization, $\boldsymbol{m}_{t+1} = \sum_{i=0}^{t} E_{t,i}\boldsymbol{g}_i$.

Assumption C.3 (Finite Execution-Stream Representation)**.** For inputs $\boldsymbol{a}_i$, the zero-start modeled update is $\widehat{\boldsymbol{u}}_t = \sum_{i=0}^{t} E_{t,i}\boldsymbol{a}_i$. For AdEMAMix the aggregate coefficient is $E_{t,i} = E_{t,i}^{(f)} + \alpha_t E_{t,i}^{(s)}$. Equality with the actual update is an optimizer-specific premise.

Proof of the buffer unrolling in Definition C.2. For generic retentions ρ_r, iterate $\boldsymbol{m}_{r+1} = \rho_r\boldsymbol{m}_r + (1-\rho_r)\boldsymbol{g}_r$. The constant choice $\rho_r = \beta^{(f)}$ recovers the AdamW first-moment recurrence:

$$\begin{aligned}
\boldsymbol{m}_{t+1} &= \rho_t\boldsymbol{m}_t + (1-\rho_t)\boldsymbol{g}_t \\
&= \rho_t\rho_{t-1}\boldsymbol{m}_{t-1} + \rho_t(1-\rho_{t-1})\boldsymbol{g}_{t-1} + (1-\rho_t)\boldsymbol{g}_t \\
&= \rho_t\rho_{t-1}\rho_{t-2}\boldsymbol{m}_{t-2} + \rho_t\rho_{t-1}(1-\rho_{t-2})\boldsymbol{g}_{t-2} \\
&\quad + \rho_t(1-\rho_{t-1})\boldsymbol{g}_{t-1} + (1-\rho_t)\boldsymbol{g}_t.
\end{aligned} \tag{C.11}$$

Each substitution replaces the remaining state $\boldsymbol{m}_{t-j}$ by $\rho_{t-j-1}\boldsymbol{m}_{t-j-1} + (1-\rho_{t-j-1})\boldsymbol{g}_{t-j-1}$, peeling off one more gradient and extending the leading ρ-product by one factor: if after m substitutions the state term reads $\big(\prod_{r=t-m+1}^{t}\rho_r\big)\boldsymbol{m}_{t-m+1}$, substituting $\boldsymbol{m}_{t-m+1} = \rho_{t-m}\boldsymbol{m}_{t-m} + (1-\rho_{t-m})\boldsymbol{g}_{t-m}$ multiplies the product by ρ_{t-m} and appends the term $\big(\prod_{r=t-m+1}^{t}\rho_r\big)(1-\rho_{t-m})\boldsymbol{g}_{t-m}$, which is exactly the $i = t-m$ term of the sum below; by induction, after $m+1$ substitutions, where the oldest exposed gradient is $\boldsymbol{g}_{t-m}$,

$$\boldsymbol{m}_{t+1} = \left(\prod_{r=t-m}^{t}\rho_r\right)\boldsymbol{m}_{t-m} \;+\; \sum_{i=t-m}^{t}(1-\rho_i)\left(\prod_{r=i+1}^{t}\rho_r\right)\boldsymbol{g}_i. \tag{C.12}$$

The convention is that an empty product is one, so the newest gradient $\boldsymbol{g}_t$ has coefficient $1-\rho_t$. Taking $m = t$ sets the oldest exposed gradient to $\boldsymbol{g}_0$ and the state term to $\big(\prod_{r=0}^{t}\rho_r\big)\boldsymbol{m}_0$, which vanishes since $\boldsymbol{m}_0 = \mathbf{0}$; this leaves $\boldsymbol{m}_{t+1} = \sum_{i=0}^{t} E_{t,i}\boldsymbol{g}_i$ with the execution kernel $E_{t,i} = (1-\rho_i)\prod_{r=i+1}^{t}\rho_r$ of (C.10). □

The execution kernel records only when a gradient input reaches future buffers, before learning rate and weight decay are applied. A slow branch (larger β) has both a small newest-step coefficient $(1-\beta)$ and a long left tail of width $O(\tau(\beta))$, and the startup transient from $\boldsymbol{m}_0 = \mathbf{0}$ fades within $O(\tau(\beta))$ steps.

Optimizer Status of the Representation. For raw SGD2MW, the raw-gradient input gives exact representation equality:

$$\boldsymbol{a}_i = \boldsymbol{g}_i, \qquad \widehat{\boldsymbol{u}}_t = \boldsymbol{u}_t.$$

For ADAMW and ADEMAMIX, the injection-time choice

$$\boldsymbol{a}_i := \frac{\boldsymbol{g}_i}{\sqrt{\boldsymbol{v}_{i+1}} + \varepsilon}$$

freezes each divisor when the input enters the buffer and defines a modeled output. The live optimizers divide the buffered numerator by the current $\sqrt{\boldsymbol{v}_{t+1}} + \varepsilon$, so the modeled and actual updates generally differ.

Finite-history predictions under uncorrelated inputs retain the time-dependent coefficients $E_{t,i}$ of this representation.

The kernel factors $\Gamma_e(h)$, $P[e]$, and $F_\delta[e]$ are defined in Definition 4.5.

Double-Sum Form of the Stationary Norm Factor. The same factor can be written as a sum over every ordered pair of input ages. For $0 \le \delta \le 1$, absolute summability of the kernel gives

$$\begin{aligned}\sum_{k\ge0}\sum_{n\ge0}\left|e_k e_n \delta^{|k-n|}\right| &\le \sum_{k\ge0}\sum_{n\ge0}|e_k||e_n| \\ &= \left(\sum_{k\ge0}|e_k|\right)^2 < \infty.\end{aligned}$$

This absolute convergence licenses splitting the double sum into its diagonal and its two strict triangles:

$$\begin{aligned}\sum_{k\ge0}\sum_{n\ge0} e_k e_n \delta^{|k-n|} &= \sum_{n\ge0} e_n^2 + 2\sum_{h\ge1}\sum_{n\ge0} e_{n+h} e_n \delta^h \\ &= P[e] + 2\sum_{h\ge1}\delta^h \Gamma_e(h) = F_\delta[e].\end{aligned}$$

The diagonal pairs have no separation and therefore carry weight one. An off-diagonal pair separated by h steps carries the weight-decay contraction factor δ^h.

Stationary Modeled Update Under Uncorrelated Inputs. Let $(\boldsymbol{a}_t)_{t\in\mathbb{Z}}$ be a two-sided input sequence satisfying Assumption 4.4, so $\mathbb{E}\boldsymbol{a}_t = \mathbf{0}$ and $\mathbb{E}\langle \boldsymbol{a}_t, \boldsymbol{a}_r\rangle = C_a(0)\mathbf{1}_{t=r}$ with $C_a(0) < \infty$. Let e be a real age kernel with $\sum_{k\geq 0}|e_k| < \infty$. For a random vector Y, write $\|Y\|_{L^2} = (\mathbb{E}\|Y\|^2)^{1/2}$. At fixed time t, define the finite cutoff

$$\widehat{\boldsymbol{u}}_t^{(N)} = \sum_{k=0}^{N} e_k \boldsymbol{a}_{t-k}.$$

The limit below increases the input-history depth at fixed t.

Lemma C.4 (Input Correlations and Modeled Update)**.** *The cutoffs converge in mean square to a zero-mean, finite-power modeled update*

$$\widehat{\boldsymbol{u}}_t^{\infty} = \lim_{N\to\infty} \widehat{\boldsymbol{u}}_t^{(N)} = \sum_{k\geq 0} e_k \boldsymbol{a}_{t-k}. \tag{C.13}$$

Its temporal correlations satisfy, for $h \geq 0$,

$$C_{\widehat{u}}(h) := \mathbb{E}\left\langle \widehat{\boldsymbol{u}}_t^{\infty}, \widehat{\boldsymbol{u}}_{t-h}^{\infty}\right\rangle = C_a(0)\Gamma_e(h). \tag{C.14}$$

Thus its instantaneous update power is $C_a(0)P[e]$ and its scalar second-order statistics are stationary. For $0 \leq \delta < 1$,

$$\mathcal{C}_\delta[\widehat{\boldsymbol{u}}^{\infty}] := C_{\widehat{u}}(0) + 2\sum_{h\geq 1}\delta^h C_{\widehat{u}}(h). \tag{C.15}$$

Then

$$\mathcal{C}_\delta[\widehat{\boldsymbol{u}}^{\infty}] = C_a(0)F_\delta[e]. \tag{C.16}$$

If the representation equals the actual update, then

$$\mathcal{C}_\delta[\boldsymbol{u}] = C_a(0)F_\delta[e]. \tag{C.17}$$

Proof of Lemma C.4. Step 1 (construct the infinite output). Absolute summability implies square summability, since

$$\sum_{k\geq 0} e_k^2 \leq \left(\sum_{k\geq 0}|e_k|\right)^2 < \infty.$$

For $N > K$, expand the squared difference before taking its expectation:

$$\begin{aligned}\mathbb{E}\left\|\widehat{\boldsymbol{u}}_t^{(N)} - \widehat{\boldsymbol{u}}_t^{(K)}\right\|^2 &= \sum_{k=K+1}^{N}\sum_{n=K+1}^{N} e_k e_n \mathbb{E}\langle \boldsymbol{a}_{t-k}, \boldsymbol{a}_{t-n}\rangle \\ &= C_a(0)\sum_{k=K+1}^{N} e_k^2 \leq C_a(0)\sum_{k>K} e_k^2 \longrightarrow 0.\end{aligned}$$

All sums in the expansion are finite. The assumption of uncorrelated inputs removes every pair with $k \neq n$; the remaining tail tends to zero by square summability. The cutoffs are therefore Cauchy in L^2, whose completeness gives the limit in (C.13). The same expansion from $k = 0$ yields

$$\left\|\widehat{\boldsymbol{u}}_t^{(N)}\right\|_{L^2}^2 = C_a(0)\sum_{k=0}^{N} e_k^2.$$

The reverse triangle inequality gives

$$\left|\left\|\widehat{\boldsymbol{u}}_t^{(N)}\right\|_{L^2} - \|\widehat{\boldsymbol{u}}_t^{\infty}\|_{L^2}\right| \leq \left\|\widehat{\boldsymbol{u}}_t^{(N)} - \widehat{\boldsymbol{u}}_t^{\infty}\right\|_{L^2} \longrightarrow 0.$$

Consequently the limit has finite power

$$\mathbb{E}\|\widehat{\boldsymbol{u}}_t^{\infty}\|^2 = C_a(0)\sum_{k\geq 0} e_k^2 = C_a(0)P[e].$$

Each cutoff has zero mean. Cauchy–Schwarz also gives

$$\|\mathbb{E}\widehat{\boldsymbol{u}}_t^{\infty}\| = \left\|\mathbb{E}(\widehat{\boldsymbol{u}}_t^{\infty} - \widehat{\boldsymbol{u}}_t^{(N)})\right\| \le \mathbb{E}\left\|\widehat{\boldsymbol{u}}_t^{\infty} - \widehat{\boldsymbol{u}}_t^{(N)}\right\| \le \left\|\widehat{\boldsymbol{u}}_t^{\infty} - \widehat{\boldsymbol{u}}_t^{(N)}\right\|_{L^2} \longrightarrow 0,$$

so the modeled update has zero mean.

Step 2 (identify repeated input steps). Fix $h \ge 0$ and take $N \ge h$. Bilinearity gives

$$\begin{aligned}\mathbb{E}\left\langle \widehat{\boldsymbol{u}}_t^{(N)}, \widehat{\boldsymbol{u}}_{t-h}^{(N)}\right\rangle &= \sum_{k=0}^{N}\sum_{n=0}^{N} e_k e_n \mathbb{E}\left\langle \boldsymbol{a}_{t-k}, \boldsymbol{a}_{t-h-n}\right\rangle \\ &= C_a(0)\sum_{n=0}^{N-h} e_{n+h}e_n.\end{aligned}$$

A pair contributes exactly when $t-k = t-h-n$, or $k = n+h$. Both cutoff indices must be at most N, which leaves $0 \le n \le N-h$. Cauchy–Schwarz for the coefficient sequences shows that

$$\sum_{n\ge 0}|e_{n+h}e_n| \le \left(\sum_{n\ge 0} e_{n+h}^2\right)^{1/2}\left(\sum_{n\ge 0} e_n^2\right)^{1/2} \le P[e] < \infty.$$

The finite sum therefore tends to $\Gamma_e(h)$.

To pass the left-hand side to its limit, add and subtract $\mathbb{E}\left\langle \widehat{\boldsymbol{u}}_t^{\infty}, \widehat{\boldsymbol{u}}_{t-h}^{(N)}\right\rangle$. Applying Cauchy–Schwarz to the two differences gives

$$\begin{aligned}&\left|\mathbb{E}\left\langle \widehat{\boldsymbol{u}}_t^{(N)}, \widehat{\boldsymbol{u}}_{t-h}^{(N)}\right\rangle - \mathbb{E}\left\langle \widehat{\boldsymbol{u}}_t^{\infty}, \widehat{\boldsymbol{u}}_{t-h}^{\infty}\right\rangle\right| \\ &\quad\le \left\|\widehat{\boldsymbol{u}}_t^{(N)} - \widehat{\boldsymbol{u}}_t^{\infty}\right\|_{L^2}\left\|\widehat{\boldsymbol{u}}_{t-h}^{(N)}\right\|_{L^2} + \|\widehat{\boldsymbol{u}}_t^{\infty}\|_{L^2}\left\|\widehat{\boldsymbol{u}}_{t-h}^{(N)} - \widehat{\boldsymbol{u}}_{t-h}^{\infty}\right\|_{L^2} \longrightarrow 0.\end{aligned}$$

The difference norms vanish by Step 1, and the other norms are bounded by $\sqrt{C_a(0)P[e]}$. This proves (C.14). The mean is zero and the correlation depends only on h. Negative step separations follow by symmetry of the real inner product, so these are stationary scalar second-order statistics.

Step 3 (sum correlations weighted by weight decay). The coefficient bound above gives $|\Gamma_e(h)| \le P[e]$. For $0 \le \delta < 1$,

$$\sum_{h\ge 1}\delta^h|C_{\widehat{u}}(h)| \le C_a(0)P[e]\sum_{h\ge 1}\delta^h = C_a(0)P[e]\frac{\delta}{1-\delta} < \infty.$$

The correlation series is therefore absolutely convergent. Substituting the proved correlation formula term by term gives

$$\begin{aligned}\mathcal{C}_\delta[\widehat{\boldsymbol{u}}^{\infty}] &= C_a(0)\Gamma_e(0) + 2C_a(0)\sum_{h\ge 1}\delta^h\Gamma_e(h) \\ &= C_a(0)\left[P[e] + 2\sum_{h\ge 1}\delta^h\Gamma_e(h)\right] = C_a(0)F_\delta[e].\end{aligned}$$

If $\widehat{\boldsymbol{u}}_t^{\infty} = \boldsymbol{u}_t$, their inner products across different steps agree at every step separation and hence their absolutely convergent weighted correlation sums agree. This gives (C.17). □

Proof of Proposition 4.6. The uncorrelated-input identity and its value at $h = 0$ supply

$$\mathcal{C}_\delta[\widehat{\boldsymbol{u}}^{\infty}] = C_a(0)F_\delta[e], \qquad C_{\widehat{u}}(0) = C_a(0)P[e].$$

Applying Theorem 4.3 to the assumed modeled response and substituting the first identity gives the stationary balance

$$\begin{aligned}(1-\delta^2)\bar{R}^2 &= \eta^2\mathcal{C}_\delta[\widehat{\boldsymbol{u}}^{\infty}] \\ &= \eta^2 C_a(0)F_\delta[e].\end{aligned}$$

Since $1-\delta^2>0$, division by this factor gives

$$\bar{R}^2=\frac{\eta^2 C_a(0)F_\delta[e]}{1-\delta^2},$$

which is (4.6).

On the nondegenerate branch,

$$C_{\widehat{u}}(0)=C_a(0)P[e]>0.$$

If $\bar{R}^2=0$, then $\boldsymbol{x}_t=\boldsymbol{x}_{t+1}=\mathbf{0}$ almost surely. The modeled-response recursion would give

$$\mathbf{0}=\boldsymbol{x}_{t+1}-\delta\boldsymbol{x}_t=-\eta\widehat{\boldsymbol{u}}_t^{\infty}\qquad\text{almost surely,}$$

contradicting $C_{\widehat{u}}(0)>0$. Therefore $\bar{R}^2>0$. Rearranging the exact radius identity now gives

$$F_\delta[e]=\frac{(1-\delta^2)\bar{R}^2}{\eta^2 C_a(0)}>0.$$

Applying Theorem 4.3 to the modeled response and substituting its instantaneous power and total norm contribution gives

$$q_{\bar{R}}^2=(1-\delta^2)\frac{C_a(0)P[e]}{C_a(0)F_\delta[e]}=(1-\delta^2)\frac{P[e]}{F_\delta[e]}.$$

When the representation equals the actual update, the promotion identities are

$$\mathcal{C}_\delta[\boldsymbol{u}]=\mathcal{C}_\delta[\widehat{\boldsymbol{u}}^{\infty}],\qquad C_u(0)=C_{\widehat{u}}(0),$$

so the same radius-stationary pair satisfies the actual-update recursion and the modeled expressions coincide with the exact actual-update conclusions. □

Extension to a Nonzero Input Mean. Let $\mathbb{E}\boldsymbol{a}_t=\boldsymbol{\mu}$ be constant, and suppose the centered inputs $\boldsymbol{a}_t-\boldsymbol{\mu}$ have time-independent power $\sigma_a^2<\infty$ and zero inner-product correlations at distinct times. For a real absolutely summable kernel and $0<\delta<1$, the modeled update has

$$\begin{aligned}C_{\widehat{u}}(0)&=\sigma_a^2P[e]+\|\boldsymbol{\mu}\|^2\left(\sum_{k\ge0}e_k\right)^2,\\ \mathcal{C}_\delta[\widehat{\boldsymbol{u}}]&=\sigma_a^2F_\delta[e]+\frac{1+\delta}{1-\delta}\|\boldsymbol{\mu}\|^2\left(\sum_{k\ge0}e_k\right)^2.\end{aligned}\tag{C.18}$$

Proof. Decompose $\widehat{\boldsymbol{u}}_t=\sum_{k\ge0}e_k(\boldsymbol{a}_{t-k}-\boldsymbol{\mu})+\boldsymbol{\mu}\sum_{k\ge0}e_k$. The first term has zero mean and, by Lemma C.4, correlation $\sigma_a^2\Gamma_e(h)$. The deterministic second term therefore adds $\|\boldsymbol{\mu}\|^2(\sum_k e_k)^2$ to every lag correlation. Its contribution to the functional is multiplied by $1+2\sum_{h\ge1}\delta^h=(1+\delta)/(1-\delta)$, giving (C.18). □

For a radius-stationary response with finite second moment, substitution into Theorem 4.3 gives the corresponding radius and relative-update scale.

C.3 Single-Timescale Memory

Proof of Corollary 4.7. Since $0\le\beta<1$, the geometric series converges and the taps have unit gain:

$$\sum_{k\ge0}e_k^{\mathrm{AdamW}}=(1-\beta)\sum_{k\ge0}\beta^k=(1-\beta)\frac{1}{1-\beta}=1.$$

The instantaneous-power factor is another geometric series:

$$\begin{aligned}P[e^{\mathrm{AdamW}}]&=\sum_{k\ge0}(e_k^{\mathrm{AdamW}})^2=(1-\beta)^2\sum_{k\ge0}\beta^{2k}\\&=(1-\beta)^2\frac{1}{1-\beta^2}=\frac{1-\beta}{1+\beta}.\end{aligned}$$

For every step separation $h \geq 0$, the kernel overlap is

$$\begin{aligned}\Gamma_{e^{\mathrm{AdamW}}}(h) &= \sum_{n\geq 0} e_n^{\mathrm{AdamW}} e_{n+h}^{\mathrm{AdamW}} \\ &= (1-\beta)^2 \sum_{n\geq 0} \beta^n \beta^{n+h} = (1-\beta)^2 \beta^h \sum_{n\geq 0} \beta^{2n} \\ &= \frac{1-\beta}{1+\beta}\,\beta^h = P[e^{\mathrm{AdamW}}]\beta^h.\end{aligned}$$

Substituting this overlap into the definition of $F_\delta[e^{\mathrm{AdamW}}]$, and summing the remaining geometric series, gives

$$\begin{aligned}F_\delta[e^{\mathrm{AdamW}}] &= P[e^{\mathrm{AdamW}}] + 2\sum_{h\geq 1} \delta^h \Gamma_{e^{\mathrm{AdamW}}}(h) \\ &= P[e^{\mathrm{AdamW}}]\left(1 + 2\sum_{h\geq 1}(\beta\delta)^h\right) \\ &= P[e^{\mathrm{AdamW}}]\left(1 + \frac{2\beta\delta}{1-\beta\delta}\right) = P[e^{\mathrm{AdamW}}]\frac{1+\beta\delta}{1-\beta\delta} \\ &= \frac{1-\beta}{1+\beta}\frac{1+\beta\delta}{1-\beta\delta},\end{aligned}$$

which proves (4.7).

The ratio $P[e^{\mathrm{AdamW}}]/F_\delta[e^{\mathrm{AdamW}}]$ cancels the common factor:

$$\frac{P[e^{\mathrm{AdamW}}]}{F_\delta[e^{\mathrm{AdamW}}]} = \frac{P[e^{\mathrm{AdamW}}]}{P[e^{\mathrm{AdamW}}](1+\beta\delta)/(1-\beta\delta)} = \frac{1-\beta\delta}{1+\beta\delta}.$$

Differentiating the power directly gives

$$\frac{\mathrm{d}}{\mathrm{d}\beta}\frac{1-\beta}{1+\beta} = \frac{-(1+\beta)-(1-\beta)}{(1+\beta)^2} = -\frac{2}{(1+\beta)^2} < 0.$$

For $F_\delta[e^{\mathrm{AdamW}}]$, logarithmic differentiation of the positive closed form gives

$$\begin{aligned}\frac{1}{F_\delta[e^{\mathrm{AdamW}}]}\frac{\mathrm{d}F_\delta[e^{\mathrm{AdamW}}]}{\mathrm{d}\beta} &= -\frac{1}{1-\beta} - \frac{1}{1+\beta} + \frac{\delta}{1+\beta\delta} + \frac{\delta}{1-\beta\delta} \\ &= -\frac{2}{1-\beta^2} + \frac{2\delta}{1-\beta^2\delta^2} \\ &= -\frac{2(1-\delta)(1+\beta^2\delta)}{(1-\beta^2)(1-\beta^2\delta^2)}.\end{aligned}$$

Multiplying by the closed form of $F_\delta[e^{\mathrm{AdamW}}]$, and cancelling $1-\beta^2 = (1-\beta)(1+\beta)$ and $1-\beta^2\delta^2 = (1-\beta\delta)(1+\beta\delta)$, gives

$$\frac{\mathrm{d}F_\delta[e^{\mathrm{AdamW}}]}{\mathrm{d}\beta} = -\frac{2(1-\delta)(1+\beta^2\delta)}{(1+\beta)^2(1-\beta\delta)^2}.$$

Finally, differentiating the ratio gives

$$\frac{\mathrm{d}}{\mathrm{d}\beta}\frac{1-\beta\delta}{1+\beta\delta} = \frac{-\delta(1+\beta\delta)-\delta(1-\beta\delta)}{(1+\beta\delta)^2} = -\frac{2\delta}{(1+\beta\delta)^2}.$$

The displayed signs follow from the stated parameter domains. □

At $0 < \delta < 1$, increasing the sole buffer's retention lowers both $F_\delta[e^{\mathrm{AdamW}}]$ and the ratio $P[e^{\mathrm{AdamW}}]/F_\delta[e^{\mathrm{AdamW}}]$. Under the zero-mean, uncorrelated-input response assumptions, this lowers the modeled radius at equal input power while cooling the stationary moment ratio. The cancellation found in short-memory analyses follows directly from the closed form: if $1-\delta \ll 1-\beta$, then $1-\beta\delta \simeq 1-\beta$ and $1+\beta\delta \simeq 1+\beta$, so $F_\delta[e^{\mathrm{AdamW}}] \simeq 1$. Pure scalar gain multiplies both $P[e^{\mathrm{AdamW}}]$ and $F_\delta[e^{\mathrm{AdamW}}]$ by the same squared factor, leaving their ratio unchanged. Adding a distinct slow pole changes the temporal shape as well as the gain.

C.4 Two-Timescale AdEMAMix Kernel Algebra

C.4.1 Exact Pairwise Contribution

The two-channel kernel is

$$e_k^{\mathrm{AMix}} = (1-\beta^{(f)})(\beta^{(f)})^k + \alpha(1-\beta^{(s)})(\beta^{(s)})^k. \tag{C.19}$$

The definitions in Sec. 4.2 and the ordered-pair identity proved above give

$$P[e^{\mathrm{AMix}}] = \sum_{k\geq 0}(e_k^{\mathrm{AMix}})^2, \qquad F_\delta[e^{\mathrm{AMix}}] = \sum_{k,n\geq 0} e_k^{\mathrm{AMix}} e_n^{\mathrm{AMix}} \delta^{|k-n|}.$$

With $0^0 = 1$, setting $\delta = 0$ in the ordered-pair sum retains exactly the diagonal $k = n$. Hence

$$P[e^{\mathrm{AMix}}] = F_\delta[e^{\mathrm{AMix}}]\big|_{\delta=0}.$$

Insert the two-channel kernel (C.19) into the two ordered slots of $F_\delta[e^{\mathrm{AMix}}]$. Expanding the product before combining either mixed term gives

$$\begin{aligned}
F_\delta[e^{\mathrm{AMix}}] = &\sum_{k,n\geq 0}\Big[(1-\beta^{(f)})(\beta^{(f)})^k + \alpha(1-\beta^{(s)})(\beta^{(s)})^k\Big]\cdot\delta^{|k-n|} \\
&\quad\cdot\Big[(1-\beta^{(f)})(\beta^{(f)})^n + \alpha(1-\beta^{(s)})(\beta^{(s)})^n\Big] \\
= &\sum_{k,n\geq 0}(1-\beta^{(f)})(\beta^{(f)})^k\cdot(1-\beta^{(f)})(\beta^{(f)})^n\cdot\delta^{|k-n|} \\
&+\alpha\sum_{k,n\geq 0}(1-\beta^{(f)})(\beta^{(f)})^k\cdot(1-\beta^{(s)})(\beta^{(s)})^n\cdot\delta^{|k-n|} \\
&+\alpha\sum_{k,n\geq 0}(1-\beta^{(s)})(\beta^{(s)})^k\cdot(1-\beta^{(f)})(\beta^{(f)})^n\cdot\delta^{|k-n|} \\
&+\alpha^2\sum_{k,n\geq 0}(1-\beta^{(s)})(\beta^{(s)})^k\cdot(1-\beta^{(s)})(\beta^{(s)})^n\cdot\delta^{|k-n|}.
\end{aligned}$$

Here k and n are two execution ages of the same input increment, and $\delta^{|k-n|}$ weights their separation by weight-decay contraction. The four ordered sums are the fast–fast, fast–slow, slow–fast, and slow–slow contributions. Their diagonal terms, summed over all four branch combinations, give instantaneous power. Their unequal-age terms add temporal correlations weighted by weight decay to the instantaneous-power factor.

Each of the four sums has the same generic form: it pairs one normalized geometric age profile with another. The following helper evaluates that pairing for any two poles.

Lemma C.5 (Exact Weighted Pairing of Two EMA Poles)**.** *For $\beta, \beta' \in [0,1)$ and $\delta \in [0,1]$, define $\Phi_\delta(\beta,\beta') := \sum_{k,n\geq 0}(1-\beta)\beta^k(1-\beta')\beta'^n\delta^{|k-n|}$. It is positive and symmetric, with*

$$\Phi_\delta(\beta,\beta') = \frac{(1-\beta)(1-\beta')(1-\delta^2\beta\beta')}{(1-\beta\beta')(1-\delta\beta)(1-\delta\beta')}. \tag{C.20}$$

Proof of Lemma C.5. Step 1 (convergence and symmetry). Since $0 \leq \delta^{|k-n|} \leq 1$, the defining double series satisfies

$$\begin{aligned}
&\sum_{k,n\geq 0}\Big|(1-\beta)\beta^k(1-\beta')\beta'^n\delta^{|k-n|}\Big| \\
&\qquad\leq (1-\beta)\sum_{k\geq 0}\beta^k(1-\beta')\sum_{n\geq 0}\beta'^n = 1.
\end{aligned}$$

It therefore converges absolutely and may be split and reindexed. Exchanging k with n and β with β' leaves the sum unchanged, which proves symmetry.

Step 2 (the diagonal). On $k = n$, the weight-decay contraction factor equals one, so the diagonal contribution is

$$\sum_{n\geq 0}(1-\beta)(1-\beta')(\beta\beta')^n = (1-\beta)(1-\beta')\frac{1}{1-\beta\beta'}.$$

Step 3 (the triangle $k > n$). Write $k = n + h$ with $h \geq 1$. Absolute convergence justifies this reindexing, and hence

$$\begin{aligned}
&\sum_{n\geq 0}\sum_{h\geq 1}(1-\beta)\beta^{n+h}(1-\beta')\beta'^{\,n}\delta^h \\
&\quad = (1-\beta)(1-\beta')\left(\sum_{n\geq 0}(\beta\beta')^n\right)\left(\sum_{h\geq 1}(\beta\delta)^h\right) \\
&\quad = \frac{(1-\beta)(1-\beta')}{1-\beta\beta'}\frac{\beta\delta}{1-\beta\delta}.
\end{aligned}$$

Step 4 (the triangle $n > k$). Writing $n = k + h$ in the same way gives

$$\begin{aligned}
&\sum_{k\geq 0}\sum_{h\geq 1}(1-\beta)\beta^{k}(1-\beta')\beta'^{\,k+h}\delta^h \\
&\quad = \frac{(1-\beta)(1-\beta')}{1-\beta\beta'}\frac{\beta'\delta}{1-\beta'\delta}.
\end{aligned}$$

Step 5 (common denominator). Adding the diagonal and the two triangles yields

$$\Phi_\delta(\beta,\beta') = \frac{(1-\beta)(1-\beta')}{1-\beta\beta'}\left(1+\frac{\beta\delta}{1-\beta\delta}+\frac{\beta'\delta}{1-\beta'\delta}\right).$$

The numerator after putting the bracket over a common denominator is

$$\begin{aligned}
&(1-\beta\delta)(1-\beta'\delta)+\beta\delta(1-\beta'\delta)+\beta'\delta(1-\beta\delta) \\
&\quad = 1-\delta^2\beta\beta'.
\end{aligned}$$

Substitution proves (C.20). Every factor in that expression is positive on the stated parameter range, so the pairing is strictly positive.

□

Two immediate parameter evaluations will be used below. At $\delta = 0$, only equal execution ages remain paired:

$$\Phi_0(\beta,\beta') = \frac{(1-\beta)(1-\beta')}{1-\beta\beta'}.$$

At $\delta = 1$, every ordered age pair receives full credit, and the closed form gives

$$\Phi_1(\beta,\beta') = \frac{(1-\beta)(1-\beta')(1-\beta\beta')}{(1-\beta\beta')(1-\beta)(1-\beta')} = 1.$$

Apply the proved identity to the four ordered sums in the preceding expansion. In their displayed order, they are

$$\Phi_\delta(\beta^{(f)},\beta^{(f)}),\qquad \alpha\Phi_\delta(\beta^{(f)},\beta^{(s)}),\qquad \alpha\Phi_\delta(\beta^{(s)},\beta^{(f)}),\qquad \alpha^2\Phi_\delta(\beta^{(s)},\beta^{(s)}).$$

The proved symmetry merges the two mixed terms. Since $P[e^{\mathrm{AMix}}] = F_\delta[e^{\mathrm{AMix}}]\big|_{\delta=0}$, the two macroscopic quantities are therefore

$$P[e^{\mathrm{AMix}}] = \Phi_0(\beta^{(f)},\beta^{(f)}) + 2\alpha\Phi_0(\beta^{(f)},\beta^{(s)}) + \alpha^2\Phi_0(\beta^{(s)},\beta^{(s)}), \tag{C.21}$$

$$F_\delta[e^{\mathrm{AMix}}] = \Phi_\delta(\beta^{(f)},\beta^{(f)}) + 2\alpha\Phi_\delta(\beta^{(f)},\beta^{(s)}) + \alpha^2\Phi_\delta(\beta^{(s)},\beta^{(s)}). \tag{C.22}$$

Expanding the two kernel slots produces three contributions. The fast–fast pairing is the baseline; the two symmetric fast–slow pairings form the first-order α contribution; and the slow–slow pairing

enters at order α^2 and controls large amplitudes. Power $P[e^{\mathrm{AMix}}]$ evaluates each pairing at $\delta = 0$, so it credits equal-age executions. The full factor $F_\delta[e^{\mathrm{AMix}}]$ additionally includes positive relative-age separations weighted by $\delta^{|k-n|}$. The comparison below therefore depends on all three pairings, not only on the linear mixed term. Equations (C.21) and (C.22) are exact deterministic kernel identities.

Proof of Theorem 4.8. Step 1 (both kernel factors increase). The closed form (C.20) is strictly positive for every admissible pair of poles, including at $\delta = 0$. Differentiating the two exact quadratics gives

$$\frac{\mathrm{d}P[e^{\mathrm{AMix}}]}{\mathrm{d}\alpha} = 2\Phi_0(\beta^{(f)}, \beta^{(s)}) + 2\alpha\Phi_0(\beta^{(s)}, \beta^{(s)}) > 0,$$
$$\frac{\mathrm{d}F_\delta[e^{\mathrm{AMix}}]}{\mathrm{d}\alpha} = 2\Phi_\delta(\beta^{(f)}, \beta^{(s)}) + 2\alpha\Phi_\delta(\beta^{(s)}, \beta^{(s)}) > 0.$$

This proves that both $P[e^{\mathrm{AMix}}]$ and $F_\delta[e^{\mathrm{AMix}}]$ increase with the slow amplitude.

Step 2 (three pairwise comparisons). For $0 < \delta < 1$, dividing the closed form by its value at $\delta = 0$ gives

$$\frac{\Phi_\delta(\beta, \beta')}{\Phi_0(\beta, \beta')} = \frac{1 - \delta^2\beta\beta'}{(1 - \delta\beta)(1 - \delta\beta')}.$$

For the fast–fast and fast–slow pairings, direct subtraction gives

$$\begin{aligned}
&\Phi_0(\beta^{(f)}, \beta^{(s)})\Phi_\delta(\beta^{(f)}, \beta^{(f)}) - \Phi_0(\beta^{(f)}, \beta^{(f)})\Phi_\delta(\beta^{(f)}, \beta^{(s)}) \\
&\quad = \Phi_0(\beta^{(f)}, \beta^{(f)})\Phi_0(\beta^{(f)}, \beta^{(s)}) \left[\frac{1 + \delta\beta^{(f)}}{1 - \delta\beta^{(f)}} - \frac{1 - \delta^2\beta^{(f)}\beta^{(s)}}{(1 - \delta\beta^{(f)})(1 - \delta\beta^{(s)})}\right] \\
&\quad = -\frac{\delta(\beta^{(s)} - \beta^{(f)})}{(1 - \delta\beta^{(f)})(1 - \delta\beta^{(s)})}\Phi_0(\beta^{(f)}, \beta^{(f)})\Phi_0(\beta^{(f)}, \beta^{(s)}).
\end{aligned}$$

Comparing the two diagonal pairings gives

$$\begin{aligned}
&\Phi_0(\beta^{(s)}, \beta^{(s)})\Phi_\delta(\beta^{(f)}, \beta^{(f)}) - \Phi_0(\beta^{(f)}, \beta^{(f)})\Phi_\delta(\beta^{(s)}, \beta^{(s)}) \\
&\quad = \Phi_0(\beta^{(f)}, \beta^{(f)})\Phi_0(\beta^{(s)}, \beta^{(s)}) \left[\frac{1 + \delta\beta^{(f)}}{1 - \delta\beta^{(f)}} - \frac{1 + \delta\beta^{(s)}}{1 - \delta\beta^{(s)}}\right] \\
&\quad = -\frac{2\delta(\beta^{(s)} - \beta^{(f)})}{(1 - \delta\beta^{(f)})(1 - \delta\beta^{(s)})}\Phi_0(\beta^{(f)}, \beta^{(f)})\Phi_0(\beta^{(s)}, \beta^{(s)}).
\end{aligned}$$

Finally, the fast–slow and slow–slow comparison is

$$\begin{aligned}
&\Phi_0(\beta^{(s)}, \beta^{(s)})\Phi_\delta(\beta^{(f)}, \beta^{(s)}) - \Phi_0(\beta^{(f)}, \beta^{(s)})\Phi_\delta(\beta^{(s)}, \beta^{(s)}) \\
&\quad = \Phi_0(\beta^{(f)}, \beta^{(s)})\Phi_0(\beta^{(s)}, \beta^{(s)}) \left[\frac{1 - \delta^2\beta^{(f)}\beta^{(s)}}{(1 - \delta\beta^{(f)})(1 - \delta\beta^{(s)})} - \frac{1 + \delta\beta^{(s)}}{1 - \delta\beta^{(s)}}\right] \\
&\quad = -\frac{\delta(\beta^{(s)} - \beta^{(f)})}{(1 - \delta\beta^{(f)})(1 - \delta\beta^{(s)})}\Phi_0(\beta^{(f)}, \beta^{(s)})\Phi_0(\beta^{(s)}, \beta^{(s)}).
\end{aligned}$$

Every denominator and every Φ_0 factor in these expressions is positive. Their signs are therefore controlled only by $\delta(\beta^{(s)} - \beta^{(f)})$.

Step 3 (the quotient derivative). Using the derivatives from Step 1, expand the quotient numerator in powers of α:

$$\begin{aligned}
&\frac{1}{2}\left(\frac{\mathrm{d}P[e^{\mathrm{AMix}}]}{\mathrm{d}\alpha}F_\delta[e^{\mathrm{AMix}}] - P[e^{\mathrm{AMix}}]\frac{\mathrm{d}F_\delta[e^{\mathrm{AMix}}]}{\mathrm{d}\alpha}\right) \\
&= \Phi_0(\beta^{(f)}, \beta^{(s)})\Phi_\delta(\beta^{(f)}, \beta^{(f)}) - \Phi_0(\beta^{(f)}, \beta^{(f)})\Phi_\delta(\beta^{(f)}, \beta^{(s)}) \\
&\quad + \alpha\big[\Phi_0(\beta^{(s)}, \beta^{(s)})\Phi_\delta(\beta^{(f)}, \beta^{(f)}) - \Phi_0(\beta^{(f)}, \beta^{(f)})\Phi_\delta(\beta^{(s)}, \beta^{(s)})\big] \\
&\quad + \alpha^2\big[\Phi_0(\beta^{(s)}, \beta^{(s)})\Phi_\delta(\beta^{(f)}, \beta^{(s)}) - \Phi_0(\beta^{(f)}, \beta^{(s)})\Phi_\delta(\beta^{(s)}, \beta^{(s)})\big].
\end{aligned}$$

Substituting the three factorizations from Step 2 and dividing by $F_\delta[e^{\mathrm{AMix}}]^2$ yields

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\alpha}\frac{P[e^{\mathrm{AMix}}]}{F_\delta[e^{\mathrm{AMix}}]} = -\frac{2\delta(\beta^{(s)}-\beta^{(f)})}{F_\delta[e^{\mathrm{AMix}}]^2(1-\delta\beta^{(f)})(1-\delta\beta^{(s)})} \\ \cdot\Big[\Phi_0(\beta^{(f)},\beta^{(f)})\Phi_0(\beta^{(f)},\beta^{(s)}) \\ + 2\alpha\Phi_0(\beta^{(f)},\beta^{(f)})\Phi_0(\beta^{(s)},\beta^{(s)}) \\ + \alpha^2\Phi_0(\beta^{(f)},\beta^{(s)})\Phi_0(\beta^{(s)},\beta^{(s)})\Big].\end{aligned}$$

Under the ordered-pole assumptions, every factor outside the leading minus sign is positive. This proves the strict ratio decrease and completes the deterministic claim. □

The sign factor isolates the boundary and reversed-order cases. At $\delta = 0$, distinct execution ages receive zero weight and the ratio is constant in α. At coincident poles, the two branches have the same age profile, so varying α is pure scalar gain and the ratio is again constant. Reversing the pole order reverses the sign of the ratio derivative.

Assumption C.6 (Equal Input Power)**.** Across the compared kernels or amplitudes, the input power $C_a(0)$ is the same positive constant and is independent of the modeled radius.

Modeled-Response Reading. Suppose now that, for every amplitude in the comparison, the input satisfies Assumption 4.4, has positive power, and drives a finite-second-moment radius-stationary response. Substituting the exact quadratics into (4.6) gives

$$q_{\bar{R}}^2(\alpha)\big|_{\text{white model}} = (1-\delta^2)\frac{\begin{array}{c}\Phi_0(\beta^{(f)},\beta^{(f)}) + 2\alpha\Phi_0(\beta^{(f)},\beta^{(s)}) \\ + \alpha^2\Phi_0(\beta^{(s)},\beta^{(s)})\end{array}}{\begin{array}{c}\Phi_\delta(\beta^{(f)},\beta^{(f)}) + 2\alpha\Phi_\delta(\beta^{(f)},\beta^{(s)}) \\ + \alpha^2\Phi_\delta(\beta^{(s)},\beta^{(s)})\end{array}}. \tag{C.23}$$

Because $1-\delta^2 > 0$, differentiating this identity and using the theorem gives

$$\frac{\mathrm{d}}{\mathrm{d}\alpha}\, q_{\bar{R}}^2(\alpha)\big|_{\text{white model}} = (1-\delta^2)\frac{\mathrm{d}}{\mathrm{d}\alpha}\frac{P[e^{\mathrm{AMix}}]}{F_\delta[e^{\mathrm{AMix}}]} < 0. \tag{C.24}$$

This moment-ratio decrease needs no comparison of input powers. Under the additional matched-power premise of Assumption C.6, the radius identity gives

$$\frac{\mathrm{d}}{\mathrm{d}\alpha}\, \bar{R}^2(\alpha)\big|_{\text{white model}} = \frac{\eta^2 C_a(0)}{1-\delta^2}\frac{\mathrm{d}F_\delta[e^{\mathrm{AMix}}]}{\mathrm{d}\alpha} > 0.$$

The radius derivative uses matched input power; the moment-ratio derivative does not.

C.5 Weight-Decay Contraction Limits Useful Memory

We hold the slow amplitude finite and ask what happens as the slow pole approaches one. Its total mass remains α, but each tap becomes small and the memory extends beyond the weight-decay timescale $\tau_w = 1/(1-\delta)$.

Proposition C.7 (Extreme Reach Returns to Fast-Only)**.** *Fix $0 < \delta < 1$, $0 \le \beta^{(f)} < 1$, and $0 < \alpha < \infty$. As $\beta^{(s)} \to 1^-$,*

$$\begin{aligned}&P[e^{\mathrm{AMix}}] \to \Phi_0(\beta^{(f)},\beta^{(f)}), \qquad F_\delta[e^{\mathrm{AMix}}] \to \Phi_\delta(\beta^{(f)},\beta^{(f)}), \\ &\frac{P[e^{\mathrm{AMix}}]}{F_\delta[e^{\mathrm{AMix}}]} \to \frac{\Phi_0(\beta^{(f)},\beta^{(f)})}{\Phi_\delta(\beta^{(f)},\beta^{(f)})}.\end{aligned} \tag{C.25}$$

The ratio equals fast-only at coincident poles, falls for every strictly ordered finite pole, and returns to fast-only at extreme reach; continuity therefore gives at least one interior maximum of its reduction.

Proof of Proposition C.7. Step 1 (the mixed terms vanish). At $\delta = 0$,

$$\Phi_0(\beta^{(f)}, \beta^{(s)}) = \frac{(1-\beta^{(f)})(1-\beta^{(s)})}{1-\beta^{(f)}\beta^{(s)}} \longrightarrow 0 \qquad \text{as } \beta^{(s)} \to 1^-.$$

For $0 < \delta < 1$, the exact mixed pairing is

$$\Phi_\delta(\beta^{(f)}, \beta^{(s)}) = \frac{(1-\beta^{(f)})(1-\beta^{(s)})(1-\delta^2\beta^{(f)}\beta^{(s)})}{(1-\beta^{(f)}\beta^{(s)})(1-\delta\beta^{(f)})(1-\delta\beta^{(s)})}.$$

Its numerator contains $1-\beta^{(s)}$, while its denominator tends to

$$(1-\beta^{(f)})(1-\delta\beta^{(f)})(1-\delta) > 0.$$

Therefore $\Phi_\delta(\beta^{(f)}, \beta^{(s)}) \to 0$.

Step 2 (the slow–slow terms vanish). The diagonal formulas give

$$\Phi_0(\beta^{(s)}, \beta^{(s)}) = \frac{1-\beta^{(s)}}{1+\beta^{(s)}} \longrightarrow 0$$

and

$$\Phi_\delta(\beta^{(s)}, \beta^{(s)}) = \frac{1-\beta^{(s)}}{1+\beta^{(s)}} \frac{1+\delta\beta^{(s)}}{1-\delta\beta^{(s)}} \longrightarrow 0.$$

The second limit uses $1-\delta > 0$.

Step 3 (return to the $\alpha = 0$ fast-branch functionals). Substituting the four limits from Steps 1 and 2 into (C.21) and (C.22) leaves only their fast–fast terms. Since $\Phi_\delta(\beta^{(f)}, \beta^{(f)}) > 0$, passage to the quotient is valid. This proves (C.25).

Step 4 (the interior maximum). At $\beta^{(s)} = \beta^{(f)}$, both quadratics acquire the same factor:

$$P[e^{\mathrm{AMix}}] = (1+\alpha)^2\Phi_0(\beta^{(f)}, \beta^{(f)}), \qquad F_\delta[e^{\mathrm{AMix}}] = (1+\alpha)^2\Phi_\delta(\beta^{(f)}, \beta^{(f)}).$$

Their ratio is therefore the fast-only value. For every $\beta^{(f)} < \beta^{(s)} < 1$, Theorem 4.8 shows that increasing α from zero to the fixed positive value strictly decreases the ratio. The closed form for every pairing is continuous in $\beta^{(s)}$ on $[\beta^{(f)}, 1)$, and Step 3 supplies a continuous extension at 1. The decrease below the fast-only value is the kernel-ratio reduction. It is continuous, vanishes at $\beta^{(s)} = \beta^{(f)}$ and in the limit $\beta^{(s)} \to 1^-$, and is positive between them. The extreme-value theorem places at least one of its maxima in the interior. □

For every slow pole in this comparison, suppose that the input has zero mean and is uncorrelated across steps, and that its modeled update has a finite-second-moment radius-stationary response. Then $q_{\bar{R}}^2 = (1-\delta^2)P[e^{\mathrm{AMix}}]/F_\delta[e^{\mathrm{AMix}}]$, so the same interior window gives a modeled moment-ratio decrease relative to fast-only.

D Controlled Validation of the Mechanism

The tangent-noise random walk gives an exact fixed-power test of the memory model. The scale-invariant MLP tests its norm predictions with autograd gradients and a live RMS preconditioner.

D.1 Noisy-Sphere Testbed and Kernel Tests

Testbed and Inputs. At each step, draw $\boldsymbol{\xi}_t \sim \mathcal{N}(\mathbf{0}, \sigma^2\boldsymbol{I})$ independently of the past and project it onto the tangent space at $\hat{\boldsymbol{x}}_t = \boldsymbol{x}_t / \|\boldsymbol{x}_t\|$:

$$\boldsymbol{a}_t = (\boldsymbol{I} - \hat{\boldsymbol{x}}_t\hat{\boldsymbol{x}}_t^\top)\boldsymbol{\xi}_t. \tag{D.1}$$

Conditional on the past, $\boldsymbol{a}_t$ has zero mean, so inputs at distinct times are uncorrelated. The projection has rank $d-1$, giving constant input power

$$C_a(0) = \mathbb{E}[\|\boldsymbol{a}_t\|^2 \mid \boldsymbol{x}_t] = \sigma^2 \operatorname{tr}(\boldsymbol{I} - \hat{\boldsymbol{x}}_t\hat{\boldsymbol{x}}_t^\top) = (d-1)\sigma^2, \tag{D.2}$$

independent of the radius. The experiment uses these noise inputs alone.

Optimizer and Reporting. We run SGD2MW, the raw two-EMA counterpart of the ADEMAMIX update in Sec. 2.1 with decoupled weight decay and no coordinatewise RMS normalization,

$$\boldsymbol{x}_{t+1} = (1-\eta\lambda)\boldsymbol{x}_t - \eta\big(\boldsymbol{m}_{t+1}^{(f)} + \alpha \boldsymbol{m}_{t+1}^{(s)}\big).$$

The representation of Assumption C.3 is then exact with $\boldsymbol{a}_i = (\boldsymbol{I} - \hat{\boldsymbol{x}}_i\hat{\boldsymbol{x}}_i^\top)\boldsymbol{\xi}_i$, and (D.1) makes each input tangent at its own weight. Define the weight-decay timescale $\tau_w = 1/(\eta\lambda)$. The default fixed-power run uses $d = 256$, $\sigma = 1$, $\eta = 0.01$, $\lambda = 0.03$, $\beta^{(f)} = 0.9$, $T = 12\tau_w$, and $\tau_w = 1/(\eta\lambda) = 3333$.

From a unit-radius start, the squared radius climbs before settling. Each constant-configuration replica is averaged over the final 20% of optimizer steps, then five replica estimates are combined. Displayed root ratios pool replica moment estimates before taking the square root, with delete-one-replica jackknife standard errors. Whole-trajectory panels retain the fixed-time replica mean and standard error.

Radial Counterfactuals. To identify the radial mechanism across the α sweep, panel (a) of Fig. 3 compares three update rules. The **full** arm uses the update above. The **power-only** arm preserves its instantaneous length but redraws its direction with an independent uniform unit vector $\boldsymbol{w}_t$,

$$\boldsymbol{x}_{t+1} = (1-\eta\lambda)\boldsymbol{x}_t - \eta\,\|\boldsymbol{u}_t\|\,\boldsymbol{w}_t,$$

destroying the persistent directional alignment. The **radial-off** arm preserves decoupled decay and projects only the optimizer update,

$$\boldsymbol{x}_{t+1} = (1-\eta\lambda)\boldsymbol{x}_t - \eta\big(\boldsymbol{I} - \hat{\boldsymbol{x}}_t\hat{\boldsymbol{x}}_t^\top\big)\boldsymbol{u}_t. \tag{D.3}$$

The projection removes the radial cross term and radial update power, retaining tangential chord growth.

Both interventions produce smaller settled radii than the full update; power only stays above radial off.

Moment Ratio and Pure-Gain Control. The slow-channel radius increase is accompanied by a lower moment ratio. At $\beta^{(s)} = 0.99$, where the slow reach is short relative to the weight-decay timescale, the derived $q_{\bar{R}}$ and its empirical tail estimate fall as α grows and follow the exact model curve with weight decay (C.23) under the ordered pole conditions of (C.24). The same-pole $\beta^{(s)} = \beta^{(f)}$ control stays flat. Pure gain multiplies both $P[e]$ and $F_\delta[e]$ by the same squared factor, leaving their ratio unchanged. The reduction with an ordered slow pole comes from its longer memory.

D.2 SCALE-INVARIANT MLP BRIDGE

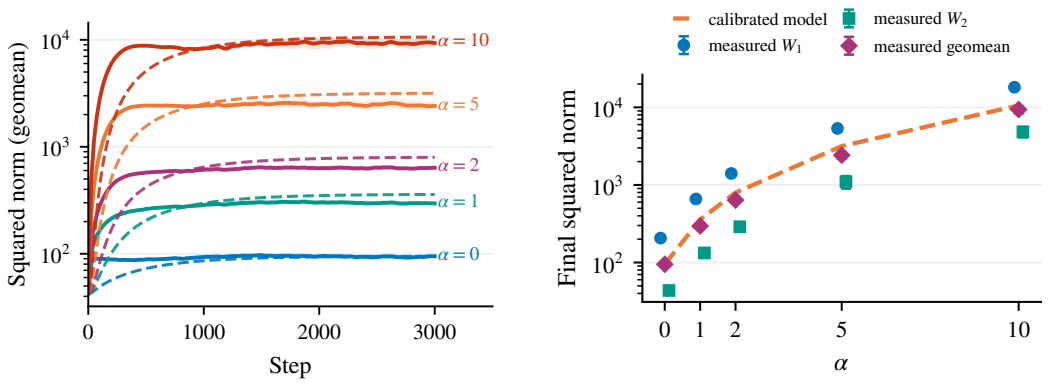
Architecture and Scale Invariance. A small MNIST classifier is trained in PyTorch with ordinary autograd gradients. The hidden blocks are bias-free linear layers followed by non-affine, epsilon-free root mean square layer normalization (RMSNorm) and a rectified linear unit (ReLU), and a fixed readout scale s_{out} sets the logit magnitude:

$$\begin{aligned}
\boldsymbol{z}_1 &= \mathrm{ReLU}(\mathrm{RMSNorm}(\boldsymbol{\Theta}_1\boldsymbol{z}_{\text{in}}))\,,\\
\boldsymbol{z}_2 &= \mathrm{ReLU}(\mathrm{RMSNorm}(\boldsymbol{\Theta}_2\boldsymbol{z}_1))\,,\\
f(\boldsymbol{z}_{\text{in}}) &= s_{\text{out}}\,\boldsymbol{\Theta}_3\boldsymbol{z}_2.
\end{aligned}$$

For a nonzero row vector $\boldsymbol{v} \in \mathbb{R}^n$, the normalization used here is

$$\mathrm{RMSNorm}(\boldsymbol{v}) = \frac{\boldsymbol{v}}{\sqrt{n^{-1}\,\|\boldsymbol{v}\|^2}},$$

with a zero-row convention at $\boldsymbol{v} = \boldsymbol{0}$. Here $\boldsymbol{z}_{\text{in}}$ is the flattened MNIST input, $\boldsymbol{z}_1, \boldsymbol{z}_2$ are hidden activations, and $\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \boldsymbol{\Theta}_3$ are the weight matrices. RMSNorm divides out positive scale, so the loss is invariant to positive scalar rescaling of either nonzero hidden matrix: $L(\kappa\boldsymbol{\Theta}_\ell) = L(\boldsymbol{\Theta}_\ell)$ for $\kappa > 0$ and $\ell \in \{1, 2\}$ (Sec. 2.2). The optimizer denominator also uses $\varepsilon = 0$, which preserves the same positive-rescaling symmetry exactly. The readout scale is fixed at $s_{\text{out}} = 4$ and never trained.



(a) (b)

Figure 6: Squared norms in the scale-invariant MNIST MLP, over five seeds. (a) Geometric mean across the two hidden matrices: solid curves show measurements with one-standard-deviation bands on the log scale, and dashed curves show the finite-history model. (b) Final norms of each matrix and their geometric mean, with one-standard-deviation bars. One input-power scale is calibrated at $\alpha = 0$ and reused for all amplitudes.

Hyperparameters and Arms. We use bias-free linear maps, hidden width 128, minibatches of 32, and 3000 steps over five seeds, at constant $\eta = 3 \times 10^{-3}$, $\lambda = 0.3$ (so $\tau_w \approx 1111$ and the horizon is $\approx 2.7\,\tau_w$), $\beta^{(f)} = 0.9$, shared RMS preconditioner with $\beta_2 = 0.99$ and $\varepsilon = 0$, and $\beta^{(s)} = 0.99$. Each seed draws its own 12,000-example training subset from the 60k MNIST pool, so the dispersion bands reflect data as well as initialization and minibatch order; within a seed the mixed and α=0 arms share the subset, keeping their ratio matched. Self-checks verify forward invariance, gradient tangency, gradient and moment rescaling, and finite outputs. Time-series bands and endpoint error bars are $\pm 1\sigma$ across seeds.

Norm Response. We compare hidden-matrix squared norms across $\alpha \in \{0, 1, 2, 5, 10\}$. The finite-radius recursion of App. E.2 starts from the observed radius. The $\alpha = 0$ endpoint calibrates one input-power scale, held fixed for every other amplitude and all plotted times. The model reproduces the ordering of the norm trajectories and their settled tails, while agreement is weaker during warmup. This tests the norm effect with live preconditioning.

E Rotational-Scheduling Details

Throughout this appendix, we use the one-based coefficient indexing $s_1, \ldots, s_T$ of the finite-horizon criterion (Anonymous authors, 2026). The main text instead indexes coefficients by the zero-based optimizer step t, so its s_t corresponds to s_{t+1} here.

E.1 The Scheduling Criterion Used in the Main Text

The criterion assumes that L is differentiable and geodesically convex on a spherical cap of angular radius $0 < \theta_{\mathrm{cap}} < \pi/4$, centered at a minimizer $\hat{\boldsymbol{x}}^\star$. Let the iterates stay in the cap, and let a fresh tangent oracle have conditional mean $\nabla_{\mathbb{S}} L(\hat{\boldsymbol{x}}_t)$ and a common second-moment bound G^2. The coefficients $s_1, \ldots, s_T$ are positive, deterministic, and fixed independently of the oracle draws. Their normalized tangent step is

$$\hat{\boldsymbol{x}}_{t+1} = \frac{\hat{\boldsymbol{x}}_t - s_t \tilde{\boldsymbol{g}}_t}{\sqrt{1 + s_t^2 \left\|\tilde{\boldsymbol{g}}_t\right\|^2}}, \qquad 1 \le t \le T. \tag{E.1}$$

The last-iterate bound at $\hat{\boldsymbol{x}}_T$ uses $T - 1$ executed updates and an auxiliary final coefficient s_T.

Write $M = \sum_{t=1}^{T} s_t$, $\omega_t = s_t/M$, and $D = \arccos\langle \hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}^\star \rangle$. The bound is

$$\mathbb{E}[L(\hat{\boldsymbol{x}}_T) - L(\hat{\boldsymbol{x}}^\star)] \le \frac{D^2}{2M} + \frac{(1+\theta_{\mathrm{cap}})G^2}{2} M C_T(\omega),$$
$$C_T(\omega) := \omega_T + \sum_{r=1}^{T-1} \frac{\omega_r^2}{\sum_{t=r+1}^{T} \omega_t}. \tag{E.2}$$

Its unique optimal positive allocation approaches linear decay on fixed bulk intervals. We use this bound as the scheduling score in Sec. 4.5.

E.2 Forward Construction: From Optimizer Memory to a Rotational Coefficient

The Inherited Input Model. Let the learning rates $\eta_t > 0$, decay coefficients $\lambda_t \ge 0$, and execution coefficients $E_{t,i}$ be deterministic. Set

$$\delta_t := 1 - \eta_t \lambda_t \in (0, 1], \qquad \widehat{\boldsymbol{u}}_t = \sum_{i=0}^{t} E_{t,i} \boldsymbol{a}_i, \qquad \boldsymbol{x}_{t+1}^{\mathrm{rep}} = \delta_t \boldsymbol{x}_t^{\mathrm{rep}} - \eta_t \widehat{\boldsymbol{u}}_t. \tag{E.3}$$

The parameter starts from fixed $\boldsymbol{x}_0 := \boldsymbol{x}_0^{\mathrm{rep}} \neq \mathbf{0}$, and the optimizer buffers start from zero. Use the zero-mean, uncorrelated inputs of Assumption 4.4, with positive power:

$$\mathbb{E}\boldsymbol{a}_i = \mathbf{0}, \qquad \mathbb{E}\langle \boldsymbol{a}_i, \boldsymbol{a}_\ell \rangle = \begin{cases} C_a(0), & i = \ell, \\ 0, & i \neq \ell, \end{cases} \qquad C_a(0) > 0. \tag{E.4}$$

These input moments determine the finite recursion (E.14), used for the MLP overlay in App. D.2 and the scheduling simulation below.

The input power $C_a(0)$ is fixed as the radius changes and matched across compared policies. This is the linear model of Sec. 4.2; the scheduling surrogate below supplies fresh tangent directions separately from the remembered updates used to compute the radius.

Momentum Coefficients and Weight Decay. Specializing the EMA coefficients of Definition C.2 to the zero-start two-buffer policies gives

$$E_{t,i} = (1-\beta^{(f)})(\beta^{(f)})^{t-i} + \alpha_t (1-\beta_i^{(s)}) \prod_{\ell=i+1}^{t} \beta_\ell^{(s)}, \qquad 0 \le i \le t. \tag{E.5}$$

The first term is the fast buffer. The second is the slow buffer, multiplied by its *execution-time* amplitude α_t, rather than the amplitude when the input was recorded. Setting $\alpha_t = 0$ gives single momentum. With constant slow coefficients the second term is $\alpha(1-\beta^{(s)})(\beta^{(s)})^{t-i}$. An empty product is one. These simulations use raw EMAs of the fixed-power inputs.

After an update is applied, weight decay contracts its displacement. We use the cumulative contraction $W_{i,t}$ defined in Definition C.1. An update applied at r enters $\boldsymbol{x}_{r+1}^{\mathrm{rep}}$, so its displacement at step t is multiplied by $W_{r+1,t}$.

Overlap and Instantaneous Power. Two modeled updates can correlate because the same input occurs in both momentum histories. Define their finite kernel inner product and its diagonal by

$$\Gamma_{t,r} := \sum_{i=0}^{\min(t,r)} E_{t,i} E_{r,i}, \qquad P_t := \Gamma_{t,t} = \sum_{i=0}^{t} E_{t,i}^2. \tag{E.6}$$

To see their meaning, expand the inner product before taking its expectation:

$$\begin{aligned} \mathbb{E}\langle \widehat{\boldsymbol{u}}_t, \widehat{\boldsymbol{u}}_r \rangle &= \sum_{i=0}^{t} \sum_{\ell=0}^{r} E_{t,i} E_{r,\ell} \mathbb{E}\langle \boldsymbol{a}_i, \boldsymbol{a}_\ell \rangle \\ &= C_a(0) \sum_{i=0}^{\min(t,r)} E_{t,i} E_{r,i} = C_a(0) \Gamma_{t,r}. \end{aligned} \tag{E.7}$$

In particular,

$$\mathbb{E}\,\|\widehat{\boldsymbol{u}}_t\|^2 = C_a(0)P_t. \tag{E.8}$$

We require $P_t > 0$ at every transferred step; the EMA policies above satisfy this condition.

Temporal Correlations in the Norm Recursion. Instantaneous power alone does not determine the radius. Define

$$R_t := \sqrt{\mathbb{E}\,\|\boldsymbol{x}_t^{\mathrm{rep}}\|^2}. \tag{E.9}$$

To identify the missing historical contribution, first unroll the parameter recursion:

$$\boldsymbol{x}_t^{\mathrm{rep}} = W_{0,t}\boldsymbol{x}_0 - \sum_{r=0}^{t-1} W_{r+1,t}\eta_r\widehat{\boldsymbol{u}}_r. \tag{E.10}$$

The fixed initialization has zero expected overlap with the centered update. Pairing the remaining terms with $\widehat{\boldsymbol{u}}_t$ and using (E.7) therefore gives

$$\mathbb{E}\,\langle \boldsymbol{x}_t^{\mathrm{rep}}, \widehat{\boldsymbol{u}}_t\rangle = -C_a(0)\sum_{r=0}^{t-1} W_{r+1,t}\eta_r\Gamma_{t,r}. \tag{E.11}$$

The sign has a physical meaning. Earlier updates displaced the parameter by $-\eta_r\widehat{\boldsymbol{u}}_r$. Positive temporal correlation aligns the current update with earlier updates, and hence against the displacement they left in the parameter. Because the applied step is $-\eta_t\widehat{\boldsymbol{u}}_t$, this negative parameter–update overlap does outward radial work on average.

Next expand one squared parameter step, as in (4.1):

$$R_{t+1}^2 = \delta_t^2 R_t^2 - 2\delta_t\eta_t\mathbb{E}\,\langle \boldsymbol{x}_t^{\mathrm{rep}}, \widehat{\boldsymbol{u}}_t\rangle + \eta_t^2\mathbb{E}\,\|\widehat{\boldsymbol{u}}_t\|^2. \tag{E.12}$$

Combining the instantaneous-power and temporal-correlation terms gives

$$F_t := P_t + 2\sum_{r=0}^{t-1}\frac{\delta_t\eta_r}{\eta_t}W_{r+1,t}\Gamma_{t,r}. \tag{E.13}$$

Here P_t is the instantaneous-power factor, and the sum accounts for temporal correlations weighted by learning rates and weight decay. The exact finite radius recursion is consequently

$$R_{t+1}^2 = \delta_t^2 R_t^2 + \eta_t^2 C_a(0)F_t. \tag{E.14}$$

Under a learning-rate schedule, F_t includes historical rate ratios as well as kernel overlap and decay; it is not a function of the execution kernel alone.

For the nonnegative EMA coefficients in (E.5), $\Gamma_{t,r} \geq 0$ and $F_t \geq P_t > 0$. The additional squared-radius contribution is then nonnegative. The identities also hold for signed execution coefficients, but an individual F_t can be negative; the positive-injection interpretation is not a claim for arbitrary signed kernels.

Accumulating the Norm Contributions. Iterating (E.14) gives the conceptual form of the radius history:

$$R_t^2 = W_{0,t}^2\,\|\boldsymbol{x}_0\|^2 + C_a(0)\sum_{r=0}^{t-1} W_{r+1,t}^2\eta_r^2 F_r. \tag{E.15}$$

The first term is the contracted initialization. Each later squared-radius contribution is contracted by subsequent weight decay. This is why the radius can keep changing after the instantaneous-power factor has almost settled. For the nonnegative EMA policies, $F_r > 0$, so the same identity gives $R_t^2 \geq W_{0,t}^2\,\|\boldsymbol{x}_0\|^2 > 0$. Thus the coefficient below has a positive radius denominator.

The Finite Relative-Update Scale. The finite-time analogue of $q_{\bar{R}}$ is $\eta_t\sqrt{C_a(0)P_t}/R_t$, with the update power and squared radius averaged before taking their root ratio.

E.2.1 From Modeled Radius to Rotational Coefficients

To isolate the scalar scheduling effect of memory, we supply the deterministic radius R_t and replace the remembered update direction by the fresh tangent oracle of App. E.1, scaled by $\sqrt{P_t}$. The oracle has its own second-moment bound G^2, as specified in App. E.1.

At optimizer step t, form the proposal along the surrogate direction $\hat{\boldsymbol{x}}_{t+1}$ and factor its positive contracted radius:

$$\begin{aligned} &\delta_t R_t \hat{\boldsymbol{x}}_{t+1} - \eta_t \sqrt{P_t} \tilde{\boldsymbol{g}}_{t+1} \\ &\quad = \delta_t R_t \left(\hat{\boldsymbol{x}}_{t+1} - \frac{\eta_t \sqrt{P_t}}{\delta_t R_t} \tilde{\boldsymbol{g}}_{t+1} \right). \end{aligned} \tag{E.16}$$

Unit normalization cancels $\delta_t R_t > 0$. Because the oracle is tangent, the remaining proposal has the normalized form (E.1) with coefficient

$$s_{t+1} = \frac{\eta_t \sqrt{P_t}}{\delta_t R_t}. \tag{E.17}$$

The single inverse-radius factor comes from normalizing the displacement. A raw scale-invariant gradient contributes an additional inverse-radius factor through (B.3).

E.3 Bound Evaluation with Prescribed Memory

We evaluate the right-hand side of (E.2) on the deterministic coefficients supplied by the finite-time model. The computation uses $T = 189{,}000$, $R_0 = 10^{-3}$, $\lambda = 0.1$, $C_a(0) = G^2 = 1$, $\theta_{\mathrm{cap}} = (1 - 10^{-9})\pi/4$, and $D = \theta_{\mathrm{cap}}/2$.

All policies use zero-start raw EMAs with $\beta^{(f)} = 0.85$. Single momentum has $\alpha_t = 0$. Fixed two-timescale memory has $\alpha_t = 12$ and $\beta_t^{(s)} = 0.9999$. Scheduled two-timescale memory uses $\alpha_t = 12(t+1)/T$ and interpolates $\beta_t^{(s)}$ from 0.85 to 0.9999 at fraction $(t+1)/T$ linearly in EMA half-life. These policies are prescribed; only the peak learning rate is selected. For each peak rate, the nominal learning rate is either constant or follows cosine decay to one percent of its peak over the full horizon, without a separate warmup. For each policy and rate family, the search evaluates 81 logarithmically spaced peaks in $[10^{-10}, 10^{-2}]$, followed by 41 logarithmically spaced peaks between the coarse neighbors of the best coarse candidate. Every candidate regenerates its own radius history and coefficients. Selection minimizes the final bound value, including the auxiliary final coefficient in (E.2).

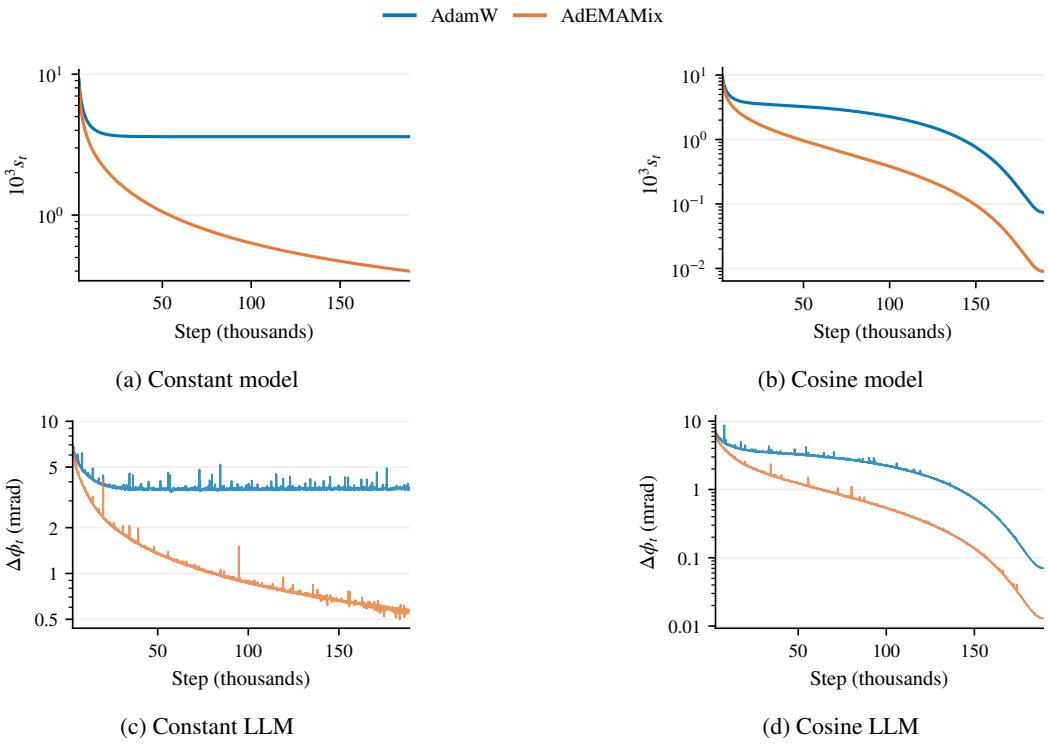
Under constant rates, both fixed and scheduled slow memory lower the best tested score. Under cosine, both raise it. These finite-grid comparisons support the qualitative expectation that endogenous cooling is more useful when explicit learning-rate decay is absent.

Settings for the Model–LLM Comparison. Fig. 7 uses the same finite model with $R_0 = 10^{-3}$, $C_a(0) = 1$, $\lambda = 0.1$, $\beta^{(f)} = 0.85$, and the LLM's scheduled slow coefficients. Both optimizer models use peak LR $8 \cdot 10^{-4}$, with the 3,000-step warmup and constant or cosine schedule of App. F.1. Curves start at step 3050. Input power is not fitted to the LLM measurements.

F LLM Experimental Details and Supplementary Results

F.1 Training, Tuning, and Measurement Protocol

Model Architecture and Pretraining Data. All pretraining experiments train a dense, decoder-only Llama-style transformer consisting of 12 transformer layers, hidden dimension $d = 768$, 12 attention heads, non-affine RMSNorm with regularizer $\epsilon = 10^{-5}$, bias-free linear projections, and tied input/output embeddings over a 50,304-token vocabulary. This configuration totals ≈ 124M parameters, of which ≈ 46M belong to non-embedding projection weights. Pretraining is performed on the FineWeb dataset tokenized at sequence length 1024 with a batch size of 64 sequences (65,536 tokens per step) across $T = 189{,}000$ steps, processing ≈ 12.4B tokens (≈ 100 tokens per parameter), corresponding to an overtrained, small-batch regime. All compared configurations share the same random seed and consume minibatches in the exact same order.



(a) Constant model

(b) Cosine model

(c) Constant LLM

(d) Cosine LLM

Figure 7: Modeled rotational coefficients (a,b) and measured LLM angles (c,d) under constant (a,c) and cosine (b,d) LR. The rows show qualitative time dependence on separate, uncalibrated scales; settings are in App. E.3.

Common Optimizer Configuration. Both AdamW and AdEMAMix share baseline retentions $\beta^{(f)} = 0.85$ and $\beta_2 = 0.999$, decoupled weight decay $\lambda = 0.1$, and a fixed global gradient-norm clipping threshold of 0.5. The nominal learning rate begins with a 3,000-step linear warmup from $\eta/100$. Standard Adam bias corrections $(1-(\beta^{(f)})^t)^{-1}$ and $(1-\beta_2^t)^{-1}$ are applied to the fast first-moment and second-moment buffers, while the slow first-moment buffer in AdEMAMix enters uncorrected (raw EMA). In the baseline AdEMAMix configuration, the slow-channel targets are $\alpha = 12$ and $\beta^{(s)} = 0.9999$, warmed over the training horizon as specified below.

Targeted Normalized Matrix Bucket. Norm interventions target five hidden-matrix classes in each of the 12 transformer blocks: the fused query-key-value projection (`qkv`), attention output (`attn_out`), and the feed-forward projections (`mlp_gate`, `mlp_up`, `mlp_down`). This defines a set of $12 \times 5 = 60$ weight-decayed matrices. Tied embeddings, unembeddings, RMSNorm affine gains, and any other no-decay parameters are excluded. Every ablation operates per-tensor independently across these 60 matrices, while all non-targeted parameters follow their native optimizer update without modification. All reported bucket metrics (such as squared radius or displacement) represent extensive sums across these 60 matrices, rather than averages of per-matrix ratios.

Learning-Rate Schedules. With warmup length $t_w = 3000$, horizon $T = 189{,}000$, and peak rate η, the warmup is

$$\eta_t = \eta \left[\frac{t}{t_w} + \left(1 - \frac{t}{t_w}\right) \frac{1}{100} \right], \qquad 0 \le t < t_w. \tag{F.1}$$

After warmup, the cosine schedule decays from its peak to one percent of the peak at step T. The flat schedule holds $\eta_t = \eta$ for $t \ge t_w$.

AdEMAMix Schedules. The slow-channel amplitude rises linearly from zero over the full horizon $T_\alpha = T$,

$$\alpha_t = \alpha \frac{t}{T_\alpha}, \qquad 0 \le t \le T_\alpha. \tag{F.2}$$

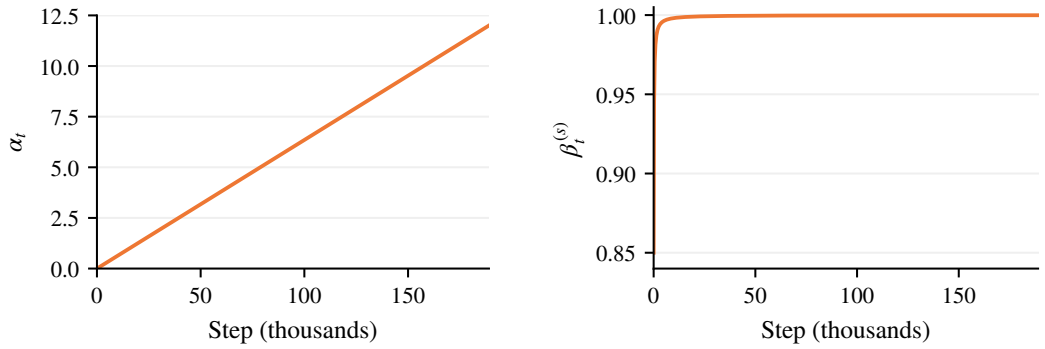
The slow retention $\beta_t^{(s)}$ is interpolated linearly in half-life. For $0 < \beta < 1$ and $\nu > -1$, set

$$\mathrm{hl}(\beta) := \frac{\ln \frac{1}{2}}{\ln \beta} - 1, \qquad \mathrm{hl}^{-1}(\nu) = \left(\frac{1}{2}\right)^{1/(\nu+1)}, \tag{F.3}$$

so that $\beta^{\mathrm{hl}(\beta)+1} = \frac{1}{2}$; the schedule is

$$\beta_t^{(s)} = \mathrm{hl}^{-1}\left(\left(1 - \frac{t}{T_\beta}\right)\mathrm{hl}(\beta_{\mathrm{start}}) + \frac{t}{T_\beta}\mathrm{hl}(\beta^{(s)})\right), \qquad 0 \le t \le T_\beta, \tag{F.4}$$

with $T_\beta = T$ and $\beta_{\mathrm{start}} = \beta^{(f)} = 0.85$. At $t = 0$ the slow buffer therefore has the fast buffer's timescale, and its half-life grows linearly to that of $\beta^{(s)} = 0.9999$. Here t denotes the scheduler's post-increment step number. The first optimizer update uses $t = 1$ in the implementation; with zero-based update indices, substitute $t + 1$ in the two warmup formulas. The difference is one step of the 189,000-step warmup.



(a) (b)

Figure 8: AdEMAMix schedules for (a) slow amplitude and (b) slow retention. Retention is interpolated in EMA half-life.

Hyperparameter Tuning Sweeps. Tab. 1 separates joint searches from conditional sweeps and fixed comparisons. Bold entries mark the selected configurations; each search uses one seed and the final validation checkpoint.

The selected cosine configurations have final validation losses 3.1527 (AdamW) and 3.1238 (AdEMAMix). AdamH's selected configuration has loss 3.1390. The AdEMAMixH amplitude comparison kept $\eta_H = 0.005$ fixed.

The constant-rate comparisons use the common grid in Tab. 2, after the 3,000-step warmup, with $\lambda = 0.1$, $\beta^{(f)} = 0.85$, and $\beta_2 = 0.999$ fixed. Fig. 9 reports the final-checkpoint losses of every arm on this grid; the radial-off arm diverged at $2 \cdot 10^{-3}$.

The tuned constant-rate losses are 3.2828 for AdamW and 3.2068 for AdEMAMix, a gap of 0.0760; the cosine gap is 0.0289.

F.2 Norm, Rotation, and Instantaneous-Power Measurements

The aggregate norm in Fig. 10 is $\|\boldsymbol{x}_t\| = (\sum_m \|\boldsymbol{x}_{m,t}\|_F^2)^{1/2}$. Norms and inner products are summed across the targeted matrices before computing the relative updates of App. B.2. At matched learning rate, their endpoint ratios obey

$$\frac{\left\|\boldsymbol{u}_{t,\perp}^{\mathrm{AMix}}\right\|}{\left\|\boldsymbol{u}_{t,\perp}^{\mathrm{AdamW}}\right\|} = \frac{q_t^{\perp,\mathrm{AMix}}}{q_t^{\perp,\mathrm{AdamW}}} \frac{\left\|\boldsymbol{x}_t^{\mathrm{AMix}}\right\|}{\left\|\boldsymbol{x}_t^{\mathrm{AdamW}}\right\|}. \tag{F.5}$$

Table 1: Search grids for cosine schedules. Bold entries form the selected configuration in each optimizer family. Conditional sweeps fix the parameters indicated in the table; extensions are additional checks.

Search or comparison	Grid and selected setting
AdamW: joint	$\beta^{(f)} \in \{\mathbf{0.85}, 0.9, 0.95\}$; $\beta_2 \in \{0.99, \mathbf{0.999}, 0.9999\}$; peak $\eta \in \{1, 3, 5, \mathbf{8}\} \cdot 10^{-4}$, at $\lambda = 0.1$
AdamW: decay	$\lambda \in \{0.005, 0.01, \mathbf{0.1}, 0.5\}$; selected $(\beta^{(f)}, \beta_2, \eta)$ fixed
AdEMAMix: slow channel	$\alpha \in \{4, 6, 8, 10, \mathbf{12}, 15\}$; $\beta^{(s)} \in \{0.99, 0.999, \mathbf{0.9999}, 0.99999\}$; $\beta^{(f)} = 0.85$, $\beta_2 = 0.999$, peak $\eta = 8 \cdot 10^{-4}$ and $\lambda = 0.1$ fixed
AdEMAMix: decay	$\lambda \in \{0.005, 0.01, \mathbf{0.1}, 0.5\}$; selected optimizer settings fixed
AdamH, joint	$\beta^{(f)} \in \{0.8, 0.85, \mathbf{0.9}\}$; $\beta_2 \in \{0.99, \mathbf{0.999}\}$; $\eta_H \in \{\mathbf{0.005}, 0.008, 0.013, 0.02\}$
AdEMAMixH, amplitude	$\alpha \in \{1, \mathbf{2}, 3, 4, 6, 8, 10, 12, 15\}$ at $\beta^{(s)} = 0.9999$ and fixed AdamH settings
AdamW/AdEMAMix, higher-rate checks	Peak $\eta \in \{2, 5\} \cdot 10^{-3}$; selected momentum and decay settings fixed
AdamH, low-rate extension	$\eta_H \in \{0.0008, 0.0013, 0.002, 0.003\}$

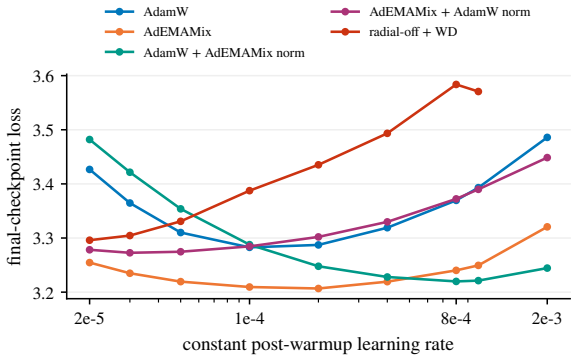


Figure 9: Final-checkpoint loss across constant rates for native and norm-transplanted arms. The minima of the two native and two transplanted arms are the runs shown in Fig. 1b; the divergent radial-off run at $2 \cdot 10^{-3}$ is omitted.

Table 2: Constant-rate search grid and arm-specific selected rates (bold). All arms use the same nine rates.

Common grid	$\{2, 3, 5\} \cdot 10^{-5} \cup \{1, 2, 4, 8\} \cdot 10^{-4} \cup \{1, 2\} \cdot 10^{-3}$
Arm	Selected rate
AdamW	$\mathbf{10^{-4}}$
AdEMAMix	$\mathbf{2 \cdot 10^{-4}}$
AdamW with AdEMAMix norms	$\mathbf{8 \cdot 10^{-4}}$
AdEMAMix with AdamW norms	$\mathbf{3 \cdot 10^{-5}}$
Radial-off	$\mathbf{2 \cdot 10^{-5}}$

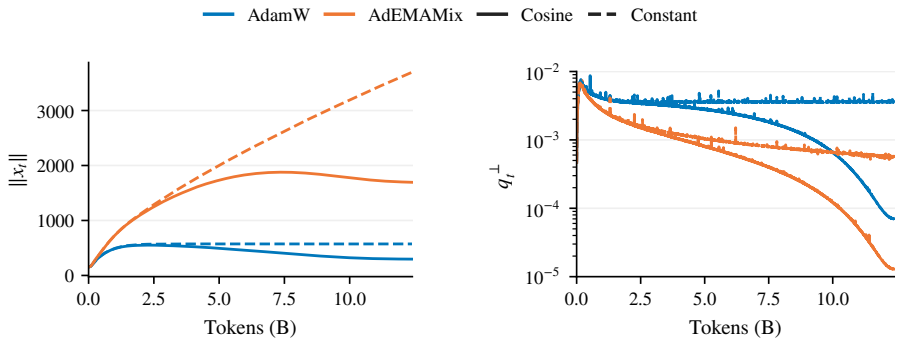
For the cosine recorder pair, this gives $0.184\sqrt{32.6} \simeq 1.05$: similar tangential displacement lengths accompany the much larger AdEMAMix radius. The endpoint squared-norm ratio under constant LR is 41.6.

The power shares use separate coordinate-recording runs with matching model, optimizer, seed, and schedule configurations. Four panels of 128 coordinates per matrix estimate the three terms in (3.1), aggregated across the 60 matrices over the final 30% of training. The recorded branches use the common live divisor, the fast-buffer bias correction, and the scheduled slow amplitude; the shares already include the factors of α_t in (3.1).

The same panels also split the survived update displacement $\sum_{t=0}^{T-1} W_{t+1,T}\eta_t \boldsymbol{u}_t$ of (C.8) at $T = 189{,}000$. Substituting $\boldsymbol{u}_t = \boldsymbol{u}_t^{(f)} + \alpha_t \boldsymbol{u}_t^{(s)}$ and expanding its squared norm gives fast, overlap, and slow terms in the same order as (3.1); each update now carries its weight-decay survival factor. Tab. 3 compares the two splits.

Table 3: Shares of the three branch terms in present update power (final 30% of training) and in the squared survived update displacement at the endpoint. Coordinate-panel estimates summed over the 60 targeted matrices.

Schedule	Quantity	Fast	Overlap	Slow
Constant	Present power	95.0%	2.0%	3.0%
	Survived displacement	1.0%	14.4%	84.6%
Cosine	Present power	93.7%	2.2%	4.1%
	Survived displacement	1.5%	21.3%	77.2%



(a) (b)

Figure 10: Matched 124M Llama recorder runs at peak rate $8 \cdot 10^{-4}$. (a) Aggregate matrix norm and (b) tangential relative update under cosine (solid) and constant (dashed) learning rates.

F.3 Intervention Definitions

Post-Step Norm Transfer. For each targeted matrix m, the post-step transfer is

$$\tilde{\boldsymbol{x}}_{m,t+1} = \|\boldsymbol{x}_{m,t+1}^{\text{donor}}\| \frac{\boldsymbol{x}_{m,t+1}^{\text{opt}}}{\|\boldsymbol{x}_{m,t+1}^{\text{opt}}\|}. \tag{F.6}$$

Here $\boldsymbol{x}_{m,t+1}^{\text{opt}}$ is the recipient's native step, including decay. The donor supplies a dense post-step norm record at the same LR schedule, seed, and data order. Rescaling changes the matrix norm while retaining its direction and optimizer buffers. The logged normalized norm is $\|\boldsymbol{x}_{m,t}\|_F \,/\, \|\boldsymbol{x}_{m,0}\|_F$.

Decay-Preserving Radial Removal. Writing $\boldsymbol{u}_t = u_{t,\parallel}\hat{\boldsymbol{x}}_t + \boldsymbol{u}_{t,\perp}$, the native update rule and the decay-preserving radial-off rule are

$$\boldsymbol{x}_{t+1}^{\text{native}} = \delta_t \boldsymbol{x}_t - \eta_t(u_{t,\parallel}\hat{\boldsymbol{x}}_t + \boldsymbol{u}_{t,\perp}), \qquad \boldsymbol{x}_{t+1}^{\text{off}} = \delta_t \boldsymbol{x}_t - \eta_t \boldsymbol{u}_{t,\perp}. \tag{F.7}$$

Their pathwise squared-radius balances satisfy

$$\|\boldsymbol{x}_{t+1}^{\text{native}}\|^2 = \delta_t^2\|\boldsymbol{x}_t\|^2 - 2\delta_t\eta_t\|\boldsymbol{x}_t\|u_{t,\|} + \eta_t^2 u_{t,\|}^2 + \eta_t^2\|\boldsymbol{u}_{t,\perp}\|^2, \tag{F.8}$$

$$\|\boldsymbol{x}_{t+1}^{\text{off}}\|^2 = \delta_t^2\|\boldsymbol{x}_t\|^2 + \eta_t^2\|\boldsymbol{u}_{t,\perp}\|^2. \tag{F.9}$$

The wrapper projects the optimizer update before applying decoupled weight decay; optimizer buffers remain unchanged.

Slow-Channel Switch. We set $\alpha_t = 0$ from the cut onward, retaining the parameters and optimizer buffers. The slow buffer continues updating but contributes nothing to the step. Cuts at $50\text{k}, 100\text{k}, 150\text{k}, 170\text{k}$ are tested under cosine peak LR $8 \cdot 10^{-4}$ and constant LR 10^{-4}, each with a matched independent ADAMW baseline. If post-cut squared-radius growth contributions agreed with the baseline, the squared-radius excess would contract by $\prod_{r=t_{\text{cut}}}^{t-1}(1-\eta_r\lambda)^2$. The measured trajectory also depends on changes in update power and radial alignment.

HYPERBALL Fixed-Radius Control. The HYPERBALL optimizer (Wen et al., 2026) eliminates the radial degree of freedom by constraining each of the 60 targeted matrices to a fixed-radius hypersphere defined by its initial Frobenius norm $r_m^H = \|\boldsymbol{x}_{m,0}\|_F$. For each constrained matrix m, the update is defined by

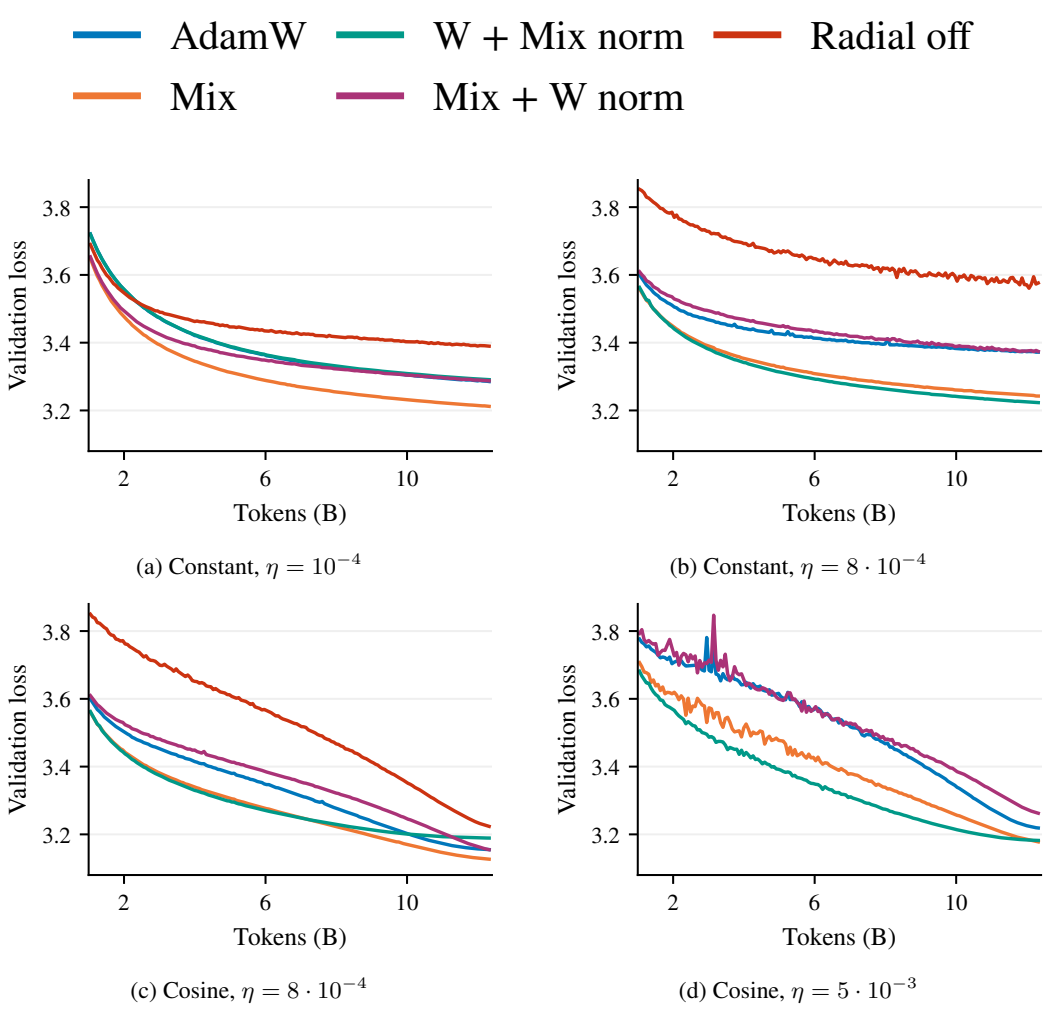
$$\boldsymbol{z}_{m,t+1} = \boldsymbol{x}_{m,t} - \eta_H r_m^H \frac{\boldsymbol{u}_{m,t}}{\|\boldsymbol{u}_{m,t}\|_F}, \tag{F.10}$$

$$\boldsymbol{x}_{m,t+1} = r_m^H \frac{\boldsymbol{z}_{m,t+1}}{\|\boldsymbol{z}_{m,t+1}\|_F}. \tag{F.11}$$

These steps normalize the proposal and retract it to the initial Frobenius radius. The step size η_H follows the cosine schedule. Constrained matrices receive no weight decay; the remaining parameters use their native ADAMW or ADEMAMIX update. The best ADEMAMIXH run reaches loss 3.1299 at $\alpha = 2$.

F.4 FULL CURVES FOR THE MAIN PERFORMANCE EXPERIMENTS

Fig. 11 shows selected absolute validation-loss trajectories at two constant and two cosine rates, spanning the operating points discussed in Sec. 5. Fig. 9 shows the full constant-rate ladder.



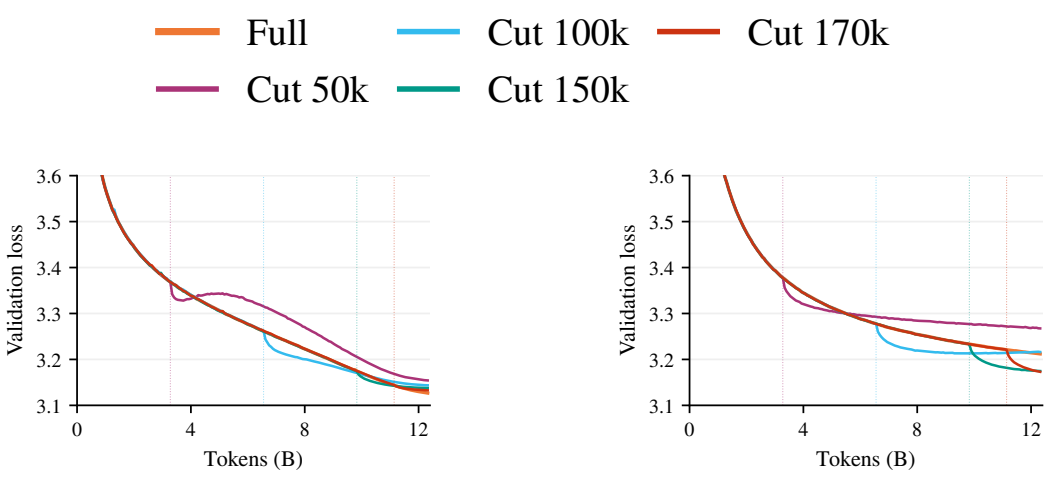
(a) Constant, $\eta = 10^{-4}$

(b) Constant, $\eta = 8 \cdot 10^{-4}$

(c) Cosine, $\eta = 8 \cdot 10^{-4}$

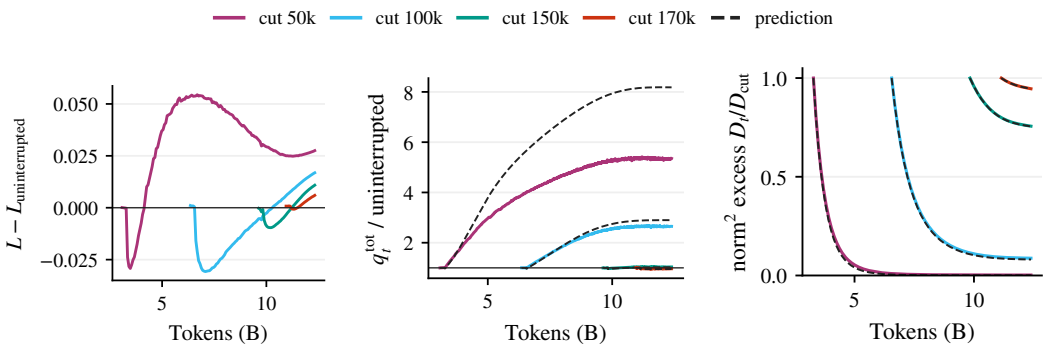
(d) Cosine, $\eta = 5 \cdot 10^{-3}$

Figure 11: Selected validation-loss trajectories at four rates, after the first billion tokens. The radial-off arm is available in the first three panels; the fourth compares native and norm-transfer runs.



(a) Cosine LR (b) Constant LR

Figure 12: Validation-loss histories after slow-channel removal at four cut times, compared with uninterrupted ADEMAMIX. Vertical markers show the cuts under cosine (left) and constant (right) LR.



(a) (b) (c)

Figure 13: The slow-channel switch under cosine LR with peak $8 \cdot 10^{-4}$; panels and predictions as in Fig. 5.

F.5 APPROXIMATE SCALE INVARIANCE OF THE MATRIX BUCKET

Finite RMSNorm regularization, residual connections, and attention scaling break exact scale invariance. Figure 15 checks two relevant properties: near-tangency of fresh gradients to the post-step weights and consistent norm growth across matrix classes. The alignment decreases during training but stays above the random-direction reference. For every class, ADEMAMIX's summed squared norm exceeds ADAMW's by more than thirtyfold at the endpoint. These diagnostics assess alignment and norm response.

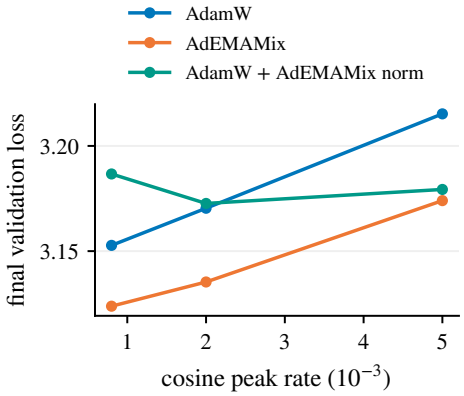


Figure 14: Final loss at three cosine peak rates for native ADAMW and ADEMAMIX and for ADAMW with ADEMAMIX's norm path.

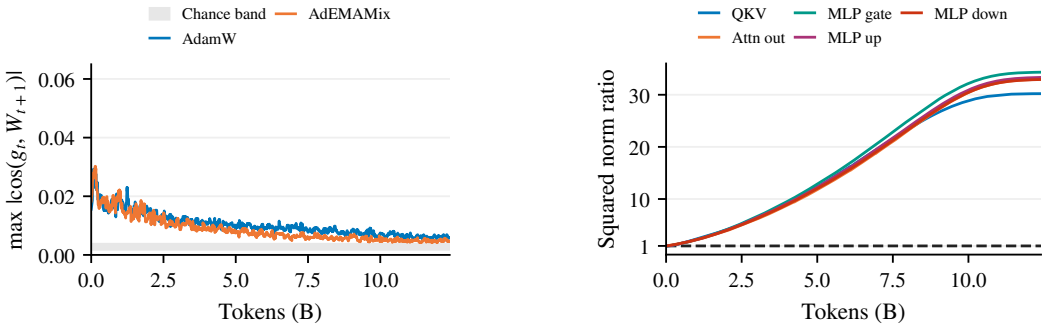


Figure 15: Gradient alignment and class-wise norm response. (a) Maximum $|\cos(\boldsymbol{g}_{m,t}, \boldsymbol{x}_{m,t+1})|$ across matrices, smoothed by a centered 21-observation median; shading gives the 5th–95th percentile for the corresponding random-direction maximum. (b) ADEMAMIX/ADAMW squared-norm ratios, summing over the 12 matrices in each class before division. The horizontal reference is one.